Universität der Bundeswehr München

Agility in Mechatronics unveiled

A value model for describing the interdependencies of agile development in mechatronics

Alexander Atzberger, M.Sc.

Vollständiger Abdruck der von der Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Gutachter:

1. Univ.-Prof. Dr.-Ing. Kristin Paetzold
   (Technische Universität Dresden)
2. Univ.-Prof. Dr.-Ing. Kilian Gericke
   (Universität Rostock)

Die Dissertation wurde am 10.06.2021 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 14.10.2021 angenommen. Die mündliche Prüfung fand am 11.11.2021 statt.

Product Development

Alexander Atzberger

Agility in Mechatronics unveiled

A value model for describing the interdependencies of agile development in mechatronics

Munich 2021

Preface

First of all, I would like to thank my dissertation advisor, Prof. Dr.-Ing. Kristin Paetzold, for providing me the opportunity to pursue this doctorate at her chair. She gave me the chance and the freedom to follow my own interests, for which I am very grateful. Furthermore, I would like to thank Prof. Dr.-Ing. Kilian Gericke, who took the time to provide me with very well-founded and valuable feedback. His support has greatly enriched this work, which I really appreciate. Both of you have been truly remarkable mentors.

Special thanks go to all those people who supported me in the context of this dissertation as interviewees or as survey participants. Without their support, this work would not have been possible, which is why I would like to express my sincerest gratitude.

In the course of this dissertation, I had the privilege of meeting a variety of very interesting people in different networks, conferences, and other events. Representing a whole range of people, I would like to gratefully acknowledge the conversations with Reiner Köttgen, Dr.-Ing. Nikola Bursac, Denis Gabriel, Jürgen Rambo and Eleonora Boffa. Furthermore, I would like to thank Jonas Heimicke and Daniel Mau as academic representatives for the outstanding cross-university collaboration and the often very refreshing videoconferences. A special thanks goes to Prof. Dr.-Ing. Dr. h.c. Sándor Vajna, whom I had the pleasure to meet during the Integrated Product Development International Summer School (ipdISS) and whose advice I highly appreciate.

I would like to thank my fellow colleagues at the Institute for Technical Product Development (ITPE) for the very pleasant working atmosphere and the stimulating discussions, especially Anne Wallisch, Joaquin Montero, Jan Mehlstäubl, Simon Nicklas, Sebastian Weber, Laura Wirths, Emir Gadzo, Martin Denk, and Julian Schönwald. On behalf of all our external doctoral students, I would like to express my gratitude to Julian Schrof, as the exchange with you was very fruitful and inspiring as well.

A huge thank you goes to my predecessor and mentor Dr.-Ing. Tobias Schmidt for being the one who recognized my inclination towards agility. I learned a great deal about agility through the lively exchange with both he and Dr.-Ing. Stefan Weiss, for which I am extremely grateful.

Last but not least, I thank the people who have always had my back and encouraged me to persevere. I thank my family and friends who have steadily supported me in my endeavor. A special thank you goes to my dear friend and proofreader from "across the pond" Andrew Switzer. And finally, I thank you, *Mau*, because you have given me so much that cannot be put into words. Thank you.

Abstract

Whereas purely material goods were once the subject of product development in a strongly product-dominated society, cyber-physical mechatronic systems and hybrid service bundles are state of the art as of today. Due to a multitude of new technologies, a strong service orientation and additional factors, this leads to a paradigm shift in value creation and thus in the development of technical products and systems. In order to exploit these potential benefits in a demand and situation-oriented manner, it is necessary to be capable of reacting quickly and adequately to changes. To remain competitive in these so-called volatile, uncertain, complex and ambiguous (VUCA) environments, agile or lightweight approaches emerged in software development at the end of the 1990s. Driven by their success in the context of software development, agility has now also found its way into the development of technical systems. Due to the fact that these agile approaches were developed in and for the context of software, the pure context transfer to physical product development leads to a number of challenges that are reflected in the understanding, interpretation, and implementation of these process models in the development of mechatronic products. Motivation

To exploit the potentials of agility in the utmost way, this thesis examines how the mechanism of action of agility can be harnessed in the context of mechatronics. The result is a model of normative elements that are essential for the agile development of mechatronic products. The model is based on two theoretical constructs: the Design for Values approach and the Taxonomy for agile development. Through this symbiosis, the model is stringently structured in itself and also provides tried-and-tested solution approaches for context-specific adaptation. Based on several preliminary studies, the problem was first validated within the scope of a quantitative investigation. The procedure for answering the research question was carried out within the framework of an embedded design approach. Its focus is qualitative in nature through two series of interviews and was supported by a final quantitative investigation. Procedure

The result is the AiM model, which identifies ten essential elements on a normative level and is based on four pillars. These normative elements are generic in nature and therefore describe which values are to be considered for the agile development of a mechatronic product or system. In addition, these are reflected against the modes of action of agile software development and classical mechatronic development to outline differences as well as similarities. Moreover, the modes of action are presented using means-end relationships to illustrate how the individual aspects interrelate in an understandable and comprehensible way. Content & Aim

The DfV approach enables the AiM model to be adapted context-specifically and thus to operationalize the model. The AiM model aims to sharpen the understanding of what characterizes agile development in the context of mechatronics. Due to its comprehensibility and simplicity, it is an aid to foster a clearer understanding of the individuals which, in turn, has a sustained impact on their mindset.

Kurzfassung

Motivation

Waren einst reine Sachgüter in einer stark produktdominierten Gesellschaft Gegenstand der Produktentwicklung, so sind heutzutage cyberphysische mechatronische Systeme sowie hybride Leistungsbündel Stand der Technik. Aufgrund einer Vielzahl an neuen Technologien, einer starken Serviceorientierung sowie weiterer Faktoren führt dies zu einem Paradigmenwechsel in der Wertschöpfung und somit der Entwicklung technischer Produkte und Systeme. Um diese Nutzenpotentiale bedarfs- und situationsgerecht heben zu können ist es notwendig, auf Veränderungen entsprechend schnell und adäquat reagieren zu können. Um weiterhin in diesen sogenannten volatilen, unsicheren, komplexen und mehrdeutigen (VUKA)-Umgebungen wettbewerbsfähig zu bleiben, haben sich agile bzw. leichtgewichtige Ansätze Ende der 90er Jahre in der Softwareentwicklung etabliert. Getrieben durch deren Erfolg im Kontext der Softwareentwicklung hat Agilität mittlerweile auch in der Entwicklung technischer Systeme Einzug gehalten. Aufgrund der Tatsache, dass diese agilen Ansätze im und für den Kontext der Software entwickelt worden sind, führt der reine Kontextübertrag in die physische Produktentwicklung zu einer Reihe von Herausforderungen, die sich im Verständnis, der Interpretation sowie der Implementierung dieser Vorgehensmodelle in der Entwicklung mechatronischer Produkte widerspiegeln.

Vorgehen

Um die Potentiale von Agilität bestmöglich ausschöpfen zu können, untersucht die vorliegende Arbeit daher, wie agile Wirkungsweisen im Kontext der Mechatronikentwicklung nutzbar gemacht werden können. Hierzu entsteht im Ergebnis ein Modell aus normativen Elementen, die für die agile Entwicklung mechatronischer Produkte essentiell sind. Das Modell basiert auf zwei theoretischen Konstrukten, dem Design for Values-Ansatz sowie der Taxonomie für die agile Entwicklung. Durch dessen Symbiose ist das Modell in sich stringent aufgebaut und liefert darüber hinaus praxiserprobte Lösungsansätze zur kontextspezifischen Adaption. Auf Basis mehrerer Vorarbeiten ist die Problemstellung zunächst im Rahmen einer quantitativen Untersuchung validiert worden. Das Vorgehen zur Beantwortung der Forschungsfrage ist im Rahmen eines Embedded Design Ansatzes erfolgt, dessen Fokus durch zwei Interviewserien qualitativer Natur ist und welcher durch eine abschließende quantitative Untersuchung untermauert wurde.

Inhalt & Zielsetzung

Im Ergebnis ist das AiM Modell entstanden, welches zehn essentielle Elemente auf normativer Ebene benennt und auf vier Grundpfeilern beruht. Diese normativen Elemente sind generischer Natur und beschreiben daher, welche Werte für die agile Entwicklung eines mechatronischen Produkts oder Systems zu berücksichtigen sind. Zudem werden diese gegenüber den Wirkungsweisen der agilen Softwareentwicklung sowie der klassischen Mechatronikentwicklung ausreflektiert, um Unterschiede als auch Ähnlichkeiten aufzuzeigen. Darüber hinaus werden die Wirkungsweisen anhand von Mittel-Zweck-Beziehungen dargelegt, um verständlich und nachvollziehbar zu beschreiben inwiefern die einzelnen Aspekte aufeinander aufbauen. Durch die Verwendung des DfV-Ansatzes ist es zudem möglich, das AiM Modell kontextspezi-

fisch zu adaptieren und somit zu operationalisieren. Das Modell zielt darauf auf, das Verständnis dahingehend zu schärfen, was agile Entwicklung im Kontext der Mechatronik charakterisiert. Aufgrund seiner Transparenz sowie seiner Einfachheit stellt es somit eine Hilfestellung dar, um das Verständnis des Individuums zu fördern und dessen Geisteshaltung (Mindset) dadurch nachhaltig zu verändern.

Contents

List of Figures

List of Tables

Acronyms

Agile Methodologies

AM	Agile Modeling
ASD	Adaptive Software Development
BDD	Behavior Driven Development
DAD	Disciplined Agile Delivery
DDD	Domain-Driven Design
DevOps	Development & Operations
FDD	Feature-Driven Design
TDD	Test-Driven Design
XP	eXtreme Programming

Agile Scaling Frameworks

LeSS	Large Scale Scrum
SAFe	Scaled Agile Framework
SoS	Scrum of Scrums

Other acronyms

AiM	Agility in Mechatronics
ASME	American Society of Mechanical Engineers
CPMS	Cyber-physical mechatronic systems
CPS	Cyber-physical systems
DACH	German-speaking countries: Germany, Austria, Switzerland
DfV	Design for Values
IoT	Internet of Things
PDCA	Plan-Do-Check-Act

PSS	Product-service systems
VDI	Association of German engineers
VSD	Value Sensitive Design

1

Introduction

"Because the design which occurs first is almost never the best possible, the prevailing system concept may need to change. Therefore, flexibility of organization is important to effective design." - Melvin Conway

Conway used this aphorism more than half a century ago to refer to flexibility as a necessary characteristic of an organization.[1] Although it was formulated under different boundary conditions in a different period, it is even more relevant today. The occurrence of change and the resulting need for organizational flexibility are two main factors that are of even greater importance nowadays.
Risk management provides to assess specific risks in advance, but ad hoc changes are difficult or hardly manageable by risk management, as it is intended to act preventively. Hence, from a company's viewpoint, certain risks can be well assessed beforehand; however, some changes are difficult to predict, yet call for a quick response. As Conway noted, a certain flexibility is needed to deal with these changes.

Agility as a solution approach

As an extension of the concept of flexibility, agility considers not only a reaction to expected and unexpected changes, but also the exploitation of those as opportunities (Böhmer et al., 2015). This altered perception that changes do not necessarily have to be perceived negatively, but as a chance for beneficial improvement, is what characterizes and thus differentiates agility.[2] In addition, agility is hallmarked by increased transparency, improved communication, and more significant commitment among team members (Schmidt et al., 2019; Atzberger et al., 2020a). Moreover, the longer a company promotes agility or strives for an agile transformation to cope with change, the more positive effects emerge over time (Schmidt et al., 2018b). Albeit, "agility is not a silver bullet" as it needs to be introduced in a sustainable manner (Atzberger et al., 2019b). The implementation causes difficulties for many companies as it cannot be carried out in a

[1] Based on these considerations, he derived a guiding principle, which is known as Conway's law: *"Organizations which design systems [...] are constrained to produce designs which are copies of the communication structures of these organizations"* (Conway, 1968, p. 31).
[2] A more in-depth discussion of the terms follows in chapter 2.2.1.

quick jolt. For agility to be beneficial, its effects and implications must be understood and adapted to the company's specific needs. In addition, the education and training of the company's employees are critical success factors for an agile transformation (Atzberger et al., 2020a). If these considerations are taken into account, agility is a very reasonable and effective means of responding to and exploiting change.

When talking about agile development, people often refer to the *Agile Manifesto*, which seventeen experienced software developers had developed in 2001. They gathered voluntarily to agree on common ground on how to develop software in a seminal and methodical manner, as a response to heavyweight, plan-driven approaches that were state of the art back then. The result of this meeting, the *Manifesto for Agile Software Development*, has been the basis for the development of many other agile process models and methodologies ever since (Beck et al., 2001). Although it does not mark the birth of agility, this event has had a significant impact on the development of software products, as agility is now state of the art in software development (VersionOne, 2020).

Agility is misinterpreted nowadays

However, the former authors of this Manifesto (who are often referred to as the "founding fathers") are at odds about the further development of agility nowadays. While some of the co-authors of the Manifesto have further developed certain methodologies and marketed them very successfully, there are others who have been concerned about the advancement of agile development since its origin. Dave Thomas is among those who are concerned about the development of agility. Thomas particularly stresses that the rationale of agility and its values "have been totally lost behind the implementation." Among several examples, he emphasizes that people tend to use agile as a noun, even spelling it with capital A – Agile. Based on this dilemma, especially the founding fathers who aim at selling their methodologies and trainings, use catchy phrases like "What is Agile?", "How to do Agile?", or "10 Ways to know Agile is cheating on you!" (Thomas, 2015). This is known as the *guru problem*[3] in scientific literature (Janes and Succi, 2012). Thomas refers to using agile as a noun as "the root of all evil", and urges people to reclaim agility (Thomas, 2015). In order to do so, people need to understand that no rules are universal as they require context (Thomas, 2015). According to Thomas, this context transfer can only be successful if the rationale of agility is understood. Therefore, he explicitly emphasizes that agility describes the ability to approach and solve problems, as *"agility is 'how' you do it, not 'what' you do"* (Thomas, 2015).

1.1 Motivation and relevance

One company that has already started to implented agile ways of working several years ago is the Swedish car and truck manufacturer Volvo. The Volvo Group has its headquarters and its most prominent car manufacturing plant on the outskirts of Gothenburg. As this company has already several years of experience with agility, transparency is one of their utmost goals. When conveying their vision, they use value-based reasoning to describe the current and future mea-

[3] The *guru problem* is explained in chapter 4.1.2 in detail.

sures. For example, the Swedish vehicle manufacturer bases its company strategy on the following values: sustainability, safety, and human-centricity. The subsequent activities of the company are derived based on these values.

Volvo's values

The first value, *sustainability*, is specified for future generations of vehicles and for Volvo's factory sites. In this context, half of all vehicles produced and sold will be fully electric by 2025. Likewise, the entire production at the Torslanda plant is to be carbon neutral by 2025. The long-term goal is to render the entire value chain carbon neutral by 2040, with all of the electricity being generated from renewable energy sources on the factory site.

The second value, *safety*, has long been operationalized by Volvo, as it was the brand that invented safety belts. Nowadays, the concept of safety considers both the occupants of the vehicle and other road users and tries to maximize safety for all involved through various technical systems such as Pilot Assist, early pedestrian detection, and frontal collision warning. All safety-relevant innovations are summarized under the term IntelliSafe.

Volvo's third value is *human-centricity* which is concretized by five cultural values focusing on both the customer and the employee. In this context, Volvo aims at high customer satisfaction by becoming the most popular manufacturer among customers. Moreover, Volvo should be the most popular employer in the entire industry for the employee according to this specification.

Although this is only a small sample, it illustrates Volvo's value-based approach and the further stringent operationalization of values into concrete goals and actions. When considering the division of Volvo Trucks, the values slightly change to environmental care, safety, and quality, as performance in terms of a high-quality product is considered to be the key to success in that context. This example shows how coherent action can be presented based on concrete values. Since the future measures are derived from these and are thus presented transparently and comprehensively, this represents a dainty form of complexity reduction due to its simplicity.

Problem description

This work aims to provide a contextual transfer of agile modes of action from software to mechatronics. According to Gericke et al. (2021), the transfer of design methodologies into practice has two particular challenges: the context-dependent adaptation of the design methodology and what they refer to as the "mindset approach".

Contextual adaptation is crucial because these process models are prescriptive in nature, therefore they need to be adapted to the context. *"In order to cover a wide range of different contexts the process models proposed in the methodologies, thus the whole design approach, became rather abstract. The high level of abstraction resulted in the perception of being of limited use"* (Gericke et al., 2021, p. 5). Therefore, a certain genericity must be present to ensure the contextual transfer. At the same time however, a concretization is needed to enable the user to adapt the abstract and context-independent approach to a specific context (Gericke et al., 2021). According to Lawson, the *"ability to manage this adaptation is one of the most important skills of designers"* (Gericke et al., 2021, p. 6).

It is precisely this ability to tailor contextually that Gericke et al. describe as "mindset approach", since *"a mindset is the proper understanding of a method's use in accordance with the designer's reality: interpretation of task, situation, execution, val-*

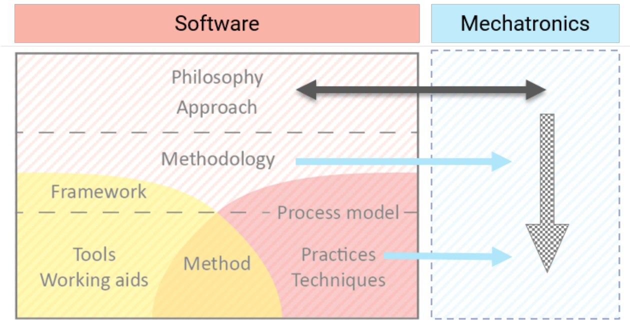


Figure 1.1
Illustration of gap and claim of work, utilizing the classification scheme by Atzberger et al. (2020a).

idation etc. and in accordance with the method's background and proper use" (Gericke et al., 2021, p. 6). This ability goes far beyond mere explanation and includes the comprehensible explanation of its modes of action. It is intended to describe the treatment of design methodologies and its prescriptive models not in a dogmatic way, but as heuristic methods that require interpretation and thus function as guidelines (Gericke et al., 2021).

Problem statement

The methods most commonly used in agile development today have been developed in and for the context of software development. The mere transfer of these methods and methodologies from the context of software to mechatronics (figure 1.1, blue arrows) hence leads to problems regarding their adaptation. To be able to apply agility in mechatronics, it is essential to consider the philosophy of agile development as a whole and translate it to the context of mechatronics (figure 1.1, black two-sided arrow) instead of only adapting the methods and methodologies to the amended context. This dissertation provides the missing context transfer, as displayed in figure 1.1.

Claim

This work claims to influence the individual's mindset, as described in the mindset approach above. The mechanisms of agility are transferred from the software context to mechatronics employing a value-based approach. In addition, assistance is provided on how these mechanisms can be operationalized in the context of mechatronics in a practical way (figure 1.1, grey arrow).

Relevance for industry

The result of non-alignment is already noticeable when investigating the current state of agile development in mechatronics (Schmidt et al., 2018b, 2019; Atzberger et al., 2020a). There are dominant methodologies and scaling frameworks, but the context-specific adaptation is a great challenge. Companies tend to adapt the existing methods at the team level (70 %) as well as in the scaled variant significantly (82 %), or they develop their own company-specific process models or combine a variety of existing scaling frameworks (Atzberger et al., 2020a). For industry, an "agile method" that tailors agile modes of action to the context of mechatronics would be most appropriate (Schmidt, 2019), taking into account that the industries in mechatronics differ significantly from each other. Therefore, it is vital to provide an aid for industry that encompasses the context of mechatronics, and that can be specified to the operational level readily.

Relevance for academia

From an academic point of view, a proper definition is needed that explains how agility is to be understood in the context of mechatronics. Since agile development is subject to the guru problem, scientifically sound answers are required to counter the ambiguities resulting from the guru problem. It has been sufficiently investigated that particular challenges exist in physical product development, and possible solutions have been identified (Atzberger and Paetzold, 2019). However, no approach exists that has gained broad acceptance in the context of physical product development so far. Furthermore, *"a holistic explanation of why agile development works does not exist"*, according to (Schmidt, 2019, p. 39). The thesis at hand addresses this gap and consequently provides insights into the mechanisms of agile product development in mechatronics for the knowledge base.

1.2 Research outline & structure of work

This work focuses on the effects of agility in the context of mechatronics. The aim is to analyze agile process models that have already been established in software development and transfer their effects. According to the classification scheme by Atzberger et al. (2020b), the effects of agile development in the form of values are anchored on the normative level. This work aims to create a transfer of the agile modes of action from agile software development to the context of mechatronics. In this regard, mechatronics is understood as physical product development, including product-service systems and cyber-physical systems.
In the following, the research question and the subordinate research questions derived from it as well as the research design is depicted briefly. The following research question can be derived from the increasing relevance of the topic:

How can the mechanisms of action of agility be harnessed in the context of mechatronics? Research question

The following three sub-research questions can be derived:
- What are the working mechanisms of agile development in mechatronics? Sub-RQ 1
- Which values are characteristic of agility in mechatronics? Sub-RQ 2
- How does agile development in mechatronics differ from agile software development? How does it differ from classical, plan-driven development? Sub-RQ 3

Each of these three sub-research questions provides a particular added value so that a comprehensive answer emerges to address the main research question. The modes of action of agility in the context of mechatronics are presented in the form of **means-end relationships**. Hence, these means-end relationships can (Research aim 1) be used to reconstruct and thus track the effects of agile development in mechatronics, as it is clear to what extent individual aspects propagate through. Next, essential elements are abstracted from these means-end relationships, which serve as fundamental normative elements of the **AiM model** and represent the (Research aim 2) underlying values of agile development in mechatronics. In a final step, the individual elements of this model are considered in relation to both agile software development and classical, plan-driven development. Thus, the **differences as** (Research aim 3) **well as commonalities** in these three perspectives become apparent and understandable. All these results are showcased within the boundaries of the model presentation and its differentiation of perspectives.

Table 1.1
Scope of dissertation.

In scope	Out of scope
Methodology-related	
✔ Development of a model	✖ Self-contained agile method or framework
✔ Alignment to company-specific values	✖ Additions to existing methods (Scrum for mechatronics)
✔ Facilitating education	✖ Change or implications to Manifesto
✔ Agility as an attribute	✖ Hybrid approaches
✔ Normative & strategic support	✖ Operative support
Company-related:	
✔ Mindset support of individuals / understanding	✖ Culture clash (agile development in classically organized companies)
✔ Strategic alignment to company-specific values / interpretation	✖ Scaling
✔ Tailoring to company strategy / implementation	✖ Team staffing
✔ Supporting agile transition	✖ Change
✔ Cultural boundary: DACH region	✖ Ecosystem
	✖ Decentralized development
Product-related:	
✔ Development of mechatronic products	✖ Development of software
✔ Cyber-physical and Product-service-systems	

Table 1.1 is presented to obtain a better overview of which aspects are examined in the context of this work and which are deliberately omitted. Here, certain criteria are specified in the areas of methodology-, company-, and product-related aspects, as it can also be found in Schmidt (2019) to identify in advance what has been considered and what has not been considered.

Structure of work

The structure of this thesis is divided into eight chapters, as displayed in figure 1.2. *Chapter 1* presents the motivation and the research questions that will be answered in this thesis.

Chapter 2 uses a funnel approach to present the state of the art, starting with the classical approaches in product development in general and moving towards agile product development. Here, the basics and chronological development are presented before focusing on agility in mechatronics in the following subchapter. Thereafter, current approaches are presented and described. Finally, the prospects for agile development are presented once again, and the work is positioned.

The research procedure is presented in detail in *chapter 3*, starting with the chosen Design science research approach, which is explained first. Next, the Design for Values approach is outlined, which, combined with the Taxonomy for agile development, forms the basis for this research. In the following, the general research procedure is explained and the methods used and the resulting insights generated are described. Afterwards, the research procedure is presented and classified again.

The description of the underlying problem is presented in *chapter 4*, where the

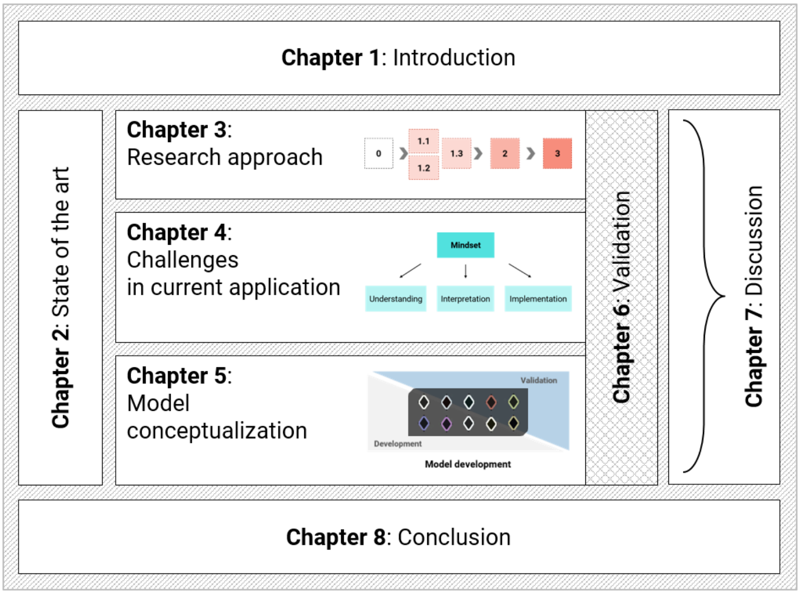


Figure 1.2
Structure of work.

topic is first operationalized into three subordinate topic areas. After explaining the topic areas, the next step is to present and discuss the chosen approach to solving the problem. Finally, the criteria that should be consulted to judge the validity of a model are listed and described.

Chapter 5 describes the development of the model successively. In the first step, the chosen perspectives and databases are depicted before the extraction of the values is illustrated in the following step. Next, the validation steps are described before the model and its values are noted. In addition, the final model is compared against the perspectives taken at the beginning of the chapter.

Chapter 6 describes the validation by checking the initial problem description before the model is validated utilizing the individual elements of the model against the criteria defined in chapter 4. Finally, the validity of the entire procedure is critically reviewed.

Chapter 7 discusses the results of this work. First, the advantages of the model for both industry and academia are presented, and it is described how well the problems outlined initially are addressed by the model. Then, strategies are presented on how the model can be applied in practice. The advantages of the model and its shortcomings are explained before the general approach is critically reviewed in a final step.

Chapter 8 summarizes the essential findings again and provides an outlook on possible further work.

2

State of the art

The state of the art is subdivided according to a so-called *funnel approach*. Such an approach aims to present an overview of the subject area initially and successively narrow it down like a funnel to the relevant subject area in the following chapters. In this way, the reader receives relevant additional information about the topic of interest and is led directly to the topic at the same time. The funnel approach for the state of the art in this thesis is visualized in figure 2.1.

In *chapter 2.1*, the basic terms and models in the context of product development are briefly explained. Furthermore, the standard approaches and methodologies, as well as the context of mechatronics, are described as well.

Chapter 2.2 explains agility in the context of product development and presents a chronicle of process models used in agile product development over time.

Chapter 2.3 describes the agile development of mechatronic systems and depicts the differences compared to agile software development. Next, the process models used in mechatronics are discussed and analyzed regarding normative elements.

Finally, *chapter 2.4* summarizes the topic of agile development. Here, the thesis presents potentials and prospects for agile development, and it shows how this work is positioned within the subject area.

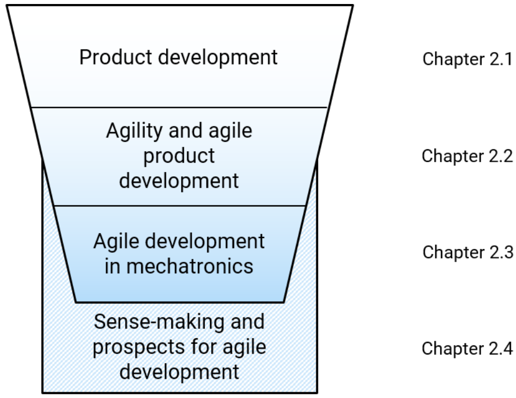


Figure 2.1
Funnel procedure detailing the state of the art.

2.1 Product development

2.1.1 Product development in general

Product development is the central setting lever representing the value chain from an initial product idea to a marketable product and is an essential component of every company (Kantelberg, 2018). At the very outset, when starting with a macro-logical view, the product life cycle of a technical product is shown in figure 2.2. According to Korthals, a **technical product** is a *"multi-part, physical product with varying degrees of complexity with regard to the functional principle, whose function is primarily based on mechanical principles, but has been realized or is supported by the extensive use of electronics, software, and (internet-based) communication and usage interfaces"* (Korthals, 2014). A more in-depth definition of a mechatronic product is presented in chapter 2.1.3.

According to figure 2.2, the **product life cycle** of a technical product is divided into various phases that a product passes through, from the initial idea to realization and use to decommissioning and recycling or further use or reuse (Bender and Gericke, 2021). This simplified representation is intended to provide a macro-logical overview of the topic.

Product creation

The semicircle colored blue in figure 2.2 represents the product creation phase. This phase is elucidated in more detail in the four-cycle model of **product and market performance development** according to Gausemeier, which is illustrated in figure 2.3. The creation of a new product or market service ranges from the product or business idea to the start of production (SOP) and includes the four areas of strategic product planning, product development, service development, and production system development (Gausemeier et al., 2018). It should be noted here that the original model according to Gausemeier and Plass comprised of three cycles and has been extended in the updated version (2018) to include the fourth cycle of service development. The foundation for this expansion is the change from pure tangible goods to hybrid service bundles. Gausemeier understands "hybrid service bundles" as a combination of a physical product, service, and business model to satisfy a comprehensive customer, provider, and user benefit (Gausemeier et al., 2018).

The first cycle *strategic product planning* aims at identifying future potential for

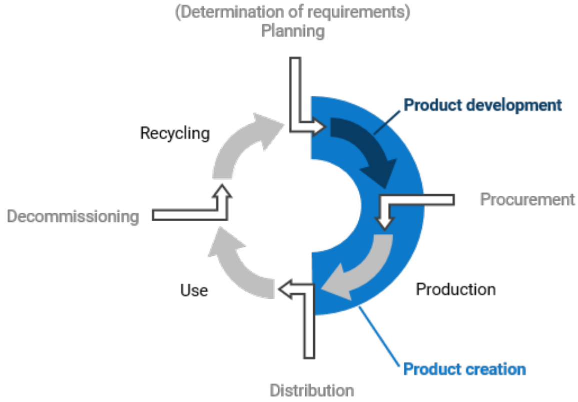


Figure 2.2
Product life cycle according to Bender and Gericke, in Lindemann (2016).

success and therefore includes the tasks of potential identification, product identification, and business planning. The second cycle, *product development*, comprises the interdisciplinary product conception, the design and the corresponding elaboration in the respective disciplines, and the integration of the results of the individual disciplines into an overall solution. Here, the product conception forms the link between the first and second cycle. The third cycle represents the transformation of a service idea into a market service. *Service development* thus includes service planning, service conception, and service integration. This cycle also involves checking whether the service adequately addresses the requirements determined in product identification. The fourth cycle of *production system development* covers operations planning, production system conception, and production system integration.

Computer-aided methods and models are increasingly used here, which is why the digital factory is often utilized. In order to achieve an efficient and innovative market performance, product, service and production system development must be carried out in close coordination with each other. Integration within cycles (such as the integration of mechanics, electronics and software in the development of mechatronic products) must be considered in the same way as cross-cycle integration, such as the use of certain manufacturing technologies. In addition, new product concepts may also require the development of manufacturing technologies and production systems. For a more thorough explanation of these four cycles see (Gausemeier et al., 2018, p.90ff.). This model is used here due to its actuality (2018) and its comprehensive overview of what market performance is considered to be nowadays. In the further course, the focus is on the second cycle *product development*.

Product development

From a systems perspective, product development describes the transformation of initial requirements from product planning as an input variable into a product model as an output variable (Lindemann, 2016). The product model can be either of physical or virtual nature, a strategy or integrated service (Lindemann, 2016), as depicted in Gausemeier's model. Product development, therefore, encompasses a wide range of activities such as the calculation and design of individual machine elements to the development of assemblies, machines or

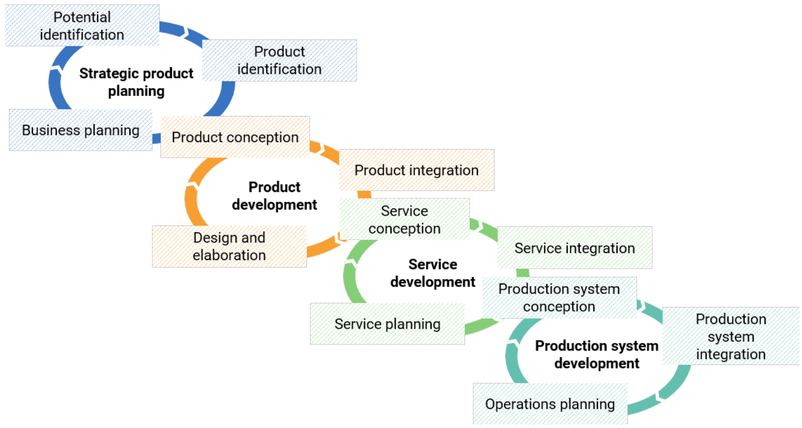


Figure 2.3
Four-cycle model according to Gausemeier et al. (2018).

systems, and the development of product-related services (Bender and Gericke, 2021). In this light, a **product development process** is defined as *"the entirety of the activities in the development of technical products as well as the control of the temporal sequence of the overall development process based on content-related focal points and the defined interfaces of the individual development phases"* (Schuh et al., 2012; Feldhusen and Grote, 2013a). The product development process can be divided into four main work phases following the intellectual logic of problem-solving: clarifying the task, designing, developing, and elaborating (Bender and Gericke, 2021).

Accompanying processes

Figure 2.4 shows an example of the main phases of a development process within a product life cycle together with the accompanying processes. VDI 2221 defines product development activities in-line with these process phases. These activities are linked to each other both logically and temporally, and in practice they can be passed through multiple times (iteratively), processed in parallel (partly), or further subdivided (decomposed) according to Bender and Gericke (2021). Product development can also be depicted as a problem-solving process, information processing, iterative process, or co-evolution of problem and solution on a rather abstract level. For a detailed explanation of these different perspectives, see Bender and Gericke (2021) and for further reading on product development activities check VDI 2221 (2019b).

Various support processes are needed along the product development cycle (and beyond), such as project management, risk management, requirements and configuration management, and several others (Ellis, 2016; Haberfellner et al., 2019; Paetzold, 2020). Since product development plays a central role in a company's value creation, there are numerous interactions with the aforementioned accompanying processes (Bender and Gericke, 2021). The accompanying processes, which are intended to support the coordination of various specialist disciplines and corporate functions, are essential in order to control and safeguard the development process (Kossiakoff, 2011; Paetzold, 2020). A detailed description of the accompanying processes can be found in Feldhusen and Grote (2013b) and Haberfellner et al. (2019).

In Gausemeier's four-cycle model, product conception is referred to as the link-

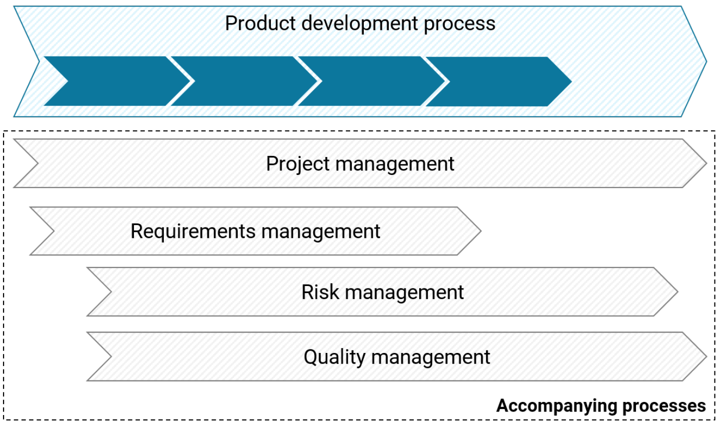


Figure 2.4 Product development process with its accompanying processes according to Paetzold (2020).

age between product planning and development. From a practical perspective, product development is a component of research and development (R&D). According to Specht's model, which can be seen in figure 2.5, fundamental research is the starting point for applied research and technology development. As a result of the research, a company can implement desired technologies into product ideas in advance development, before a product model is created in series product development. At last, the product is then handed over to production and introduced into the market (Specht et al., 2002).

In addition to Specht's model, Ehrlenspiel and Meerkamm also present similar phase-based models which, based on a previous technology project, illustrate the individual phases within a development project; on the one hand with possible prototypes using the example of car development, and on the other hand with exemplary product development methods for an industrial device (Ehrlenspiel and Meerkamm, 2017). Chapter 7.4 discusses the use and classification of agile approaches and process models in phase-based procedures as well.

Early phases

In a model by Link, the term *early phase* in the innovation process or *fuzzy front end* is used. The early phase covers all activities from the generation of ideas to the development decision (Link, 2014). The fuzzy front end has resemblances with the product planning cycle in Gausemeier's model. Ideas for new products or services can be derived from corporate strategy, a new technology, or a customer need (Link, 2014). There are, however, apparent differences between an early phase and a new product development cycle in terms of the way of working, activities, commercialization date, expected revenue, and funding. These are described in more detail in (Link, 2014, p. 73f.).

Here, the focus is on the development process of a physical product. For this reason, only this cycle and the corresponding models will be considered in the further course, whereas the other depictions should only serve as aids for classification.

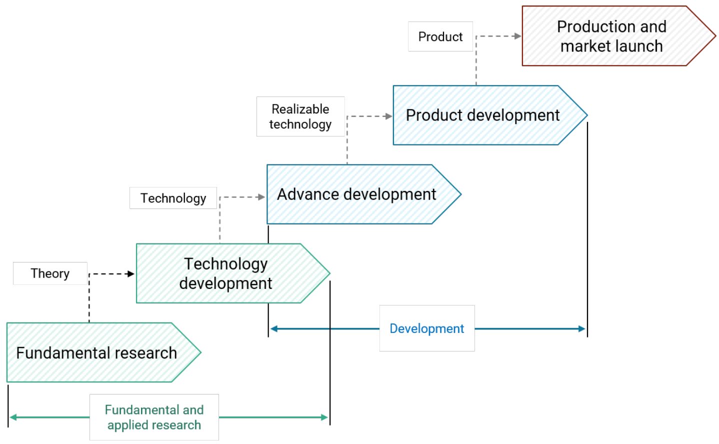


Figure 2.5
Alignment of product development into R&D according to Specht et al. (2002).

2.1.2 Current approaches and methodologies in product development

In the following section, various approaches to product development are analyzed in more detail. The most relevant methodologies are explained to provide an overview of the current state of product development process models. There are already some works available that deal with the (historical) development of process models in product development. A detailed elucidation of the advancement of process models starting in the mid-19th century (starting with Redtenbacher & Releaux) until the turn of the millennium can be found in Heymann (2005). An overview of various types of process models based on different classification schemes is depicted in Gericke and Blessing (2011), which has been examined even further in Gericke and Blessing (2012).

Wynn and Clarkson have developed and successively revised an organizing framework to classify process models in design and development (Wynn and Clarkson, 2005, 2018). The models are distinguished according to their scope into macro-, meso-, and micro-level models. Moreover, the framework comprises four model types: *procedural*, *analytical*, *abstract*, and *management science/operations research* models (Wynn and Clarkson, 2018). In the following, macro-level models according to the distinction made by Wynn and Clarkson are examined in more detail before moving on to meso-level models. The following boundary conditions limit the selection: only current procedures (2010 and later) are examined, with a focus on cross-disciplinary collaboration, as it is essential for mechatronic product development. Additionally, only models which are used in the German-speaking DACH region are included.

Approaches The following macro-level models represent established approaches in the context of product development in the DACH region. The first approach is *Integrated Product Development* (IPD). The origins can be traced back to Olsson and Andreasen and Hein in the mid-1980s. On this basis, IPD itself has developed further to its most recent representations, such as Albers et al. (2016b) and Ehrlenspiel and Meerkamm (2017), but also slight variants have been de-

IPD

veloped, such as Integrated Design Engineering (IDE) according to Vajna (2014) and Dynamic Product Development according to Ottosson (2019). A chronological presentation regarding the advancements of IPD can be found in Vajna (2014). Although these approaches differ in their characteristics, the IPD approaches have in common that they offer a solution approach to overcoming the strong division of labor in product creation (Ehrlenspiel and Meerkamm, 2017). Integrated product development can thus be understood as a holistic approach that combines the goal-oriented combination of organizational, methodological, and technical measures and tools to support the individual within the product development process as best as possible (Albers et al., 2016b; Ehrlenspiel and Meerkamm, 2017). Moreover, Meerkamm and Paetzold describe IPD as a four-leaf clover, which entails methodology, technology, organization, and the human being (Meerkamm and Paetzold, 2009).

Simultaneous engineering

Another approach (which is often combined with IPD) is process-oriented simultaneous engineering . It considers the entire product creation phase and refers to the targeted, interdisciplinary cooperation and parallel work of product, production, and sales development, supported by project management (Ehrlenspiel and Meerkamm, 2017). In their book, Ehrlenspiel and Meerkamm propose a significant reduction in production cost and processing time, which results from this workflow parallelization. A detailed description of its benefits and potential risks is presented in Ehrlenspiel and Meerkamm (2017). While simultaneous engineering overlaps different and previously consecutive activities (such as development and design), concurrent engineering divides one single task among several people, who are working on that task in parallel (Vajna, 2014). An essential aspect of both approaches is the question of when the results of the previously started work step are stable to such an extent that the probability of a change and the associated change costs are sufficiently low (Vajna, 2014). A more precise distinction between these very similar approaches is illustrated in Vajna (2014). Currently, especially in the German-speaking region, the term simultaneous engineering is used as an umbrella term, as it is the more holistic approach. However, it implies the idea of concurrent engineering, which is why the term *concurrent simultaneous engineering*, as once created by Bullinger, would be more appropriate (Bullinger and Warschat, 1996; Vajna, 2014).

Concurrent engineering

Systems engineering

Systems engineering represents an interdisciplinary approach that focuses on the design of complex technical systems instead of the individual parts within the system (Blanchard and Fabrycky, 2014; Lindemann, 2016). According to INCOSE, it provides methods and processes for the realization of successful systems, whereby the clarification of requirements, the system architecture, and the integration of subsystems and components are particularly important. This requires intensive interface consideration, which is the basis for subsequent testing, validation, and verification of the developed partial solutions and the overall system – also referred to as systems thinking. The focus is on quality from the user's perspective and the system's service life, including psychological, social, and ecological aspects (INCOSE, 2014). Schulze contrasts different definitions of the term and cites the metaphor of a *"director who controls an orchestra and thus the individual musicians and instruments"* (Lindemann, 2016, p. 153f.). A more in-depth description of systems engineering and its characteristics can be found in

Kossiakoff (2011), Blanchard and Fabrycky (2014), or Haberfellner et al. (2019). Furthermore, the organizing framework according to Wynn and Clarkson lists the Stage-Gate process at the macro-level as well. It has a strong project management character which is intended to support the engineering design process. Although Cooper's Stage-Gate model was already developed in the mid-80s (Cooper, 1986), it has gained broad attention and has established itself to this day, not least because of its numerous adaptations. Since it has been further developed in recent years into a hybrid approach with agile procedures, this is discussed in more detail in chapter 2.3.2. However, the Stage-Gate model in its classical form is still widely used in the industry as a workflow management approach for time planning (Lindemann, 2016).

Stage-Gate

These approaches are relevant, but they are too abstract for concrete application due to their macro-logic nature. Therefore, meso-level models that are neither too abstract (macro-level) nor too concrete (micro-level) are considered in the following.

Methodologies There are a number of meso-level methodologies for product development, which can be subdivided according to their respective disciplines, as presented in Gericke and Blessing (2012). Therefore, this section is limited to the same boundary conditions as before: Only the most recent methodologies (2010 and later) are considered which have been designed for the context of mechatronics and are used in German-speaking countries.

VDI 2206 (V model)

The generic V model depicted in VDI guideline 2206 has been developed in 2004 and updated in 2020 (VDI 2206, 2004, 2020), making it the most up-to-date process model today. The revision was carried out to expand the methodology to include cyber-physical aspects, thereby increasing the understanding of expansions of mechatronic systems and cyber-physical functions. The V model serves as a factual framework and orientation for the interdisciplinary development of cyber-physical mechatronic systems (CPMS). The macrocycle of the V model is divided into the steps of system design on the descending branch, domain-specific design at the bottom, and system integration and feature validation on the rising branch of the "V". By repeatedly passing through these stages, the degree of product maturity gradually increases. Due to its generic nature, the process model offers the possibility for adaptability and is used in other disciplines as well, such as systems engineering (highly complex system development) or in a tailored version as *V model XT*.

W model

According to Nattermann and Anderl, the W model is based on the V model and created by stringing together two V cycles. It represents a process model for the development of active systems, which serves the structural integration of mechatronic and adaptronic systems and, due to the high level of interconnectedness of the disciplines, proposes a continuous, cross-disciplinary synchronization. The difference to the V model becomes apparent during the development of discipline-specific components; here, the integration of digital models of the sub-discipline-specific solutions is carried out for the virtual validation of the interdisciplinary development. In addition, the use of systems engineering principles and thus, a data management system that allows for the analysis and synchronization of discipline-specific datasets, is proposed (Nattermann and Anderl, 2013).

Model-based virtual product development (MVPE) according to Eigner is the continuous, computer-supported, formal modeling and documentation alongside all phases of the product life cycle which are relevant for development. The objective is the early development of product and production knowledge and thus the early optimization of product properties in a holistic optimization of the entire product life cycle. The MVPE is divided into four main parts: interdisciplinary system development, discipline-specific detailing, system integration and property assurance, and system life cycle and product life cycle management backbone. It represents a further development of the V model (2004) as well, whereby the virtualization of product development should offer a considerable reduction of physical prototypes and result in an increase in efficiency (Eigner, 2014). MVPE

The integrated product development model (iPeM) by Albers et al. is a metamodel of product development that provides all elements necessary to derive context-specific models for product development projects. It reconciles the strategic and the operational view and enables the description of product development activities carried out in the course of the process in a parallel or iterative form. Each activity is supported in the transformation of an activity-specific actual state into the target state with the universal problem-solving methodology SPALTEN[1], which stands for seven problem-solving activities: situation analysis, problem delimitation, alternative solutions, solution selection, scope analysis, decision & implementation, and follow-up & learning. They structure the problem-solving process through a targeted alternation of information generation and condensation and presuppose the identification of the actual problem before the targeted search for a solution utilizing creativity (Albers et al., 2016a). iPeM

The VDI guideline 2221 (Methodology for the development and design of technical systems and products) was once created in 1993 and revised in 2019 (VDI 2221, 1993, 2019a,b). It is a well-established methodology and therefore serves as the most up-to-date reference point in mechatronic system development. The concept of the iPeM mentioned above has been substantially transferred to the revised version of VDI 2221. It is divided into two parts, with sheet one dealing with the basics of methodical development of all types of technical products and systems and prescribing central goals, activities, and work results in a "model of product development", which shall be understood as a guideline. In the second sheet, six case studies from different contexts and product types are shown as examples, illustrating the tailoring of the generic process model to specific contexts. The original guideline has a strong prescriptive and thus textbook character, which has become much more user-friendly in the most recent version, mainly due to the second sheet. VDI 2221

In addition to the process models and methodologies described here, current models include the four-cycle model according to Gausemeier et al., which was already shown in chapter 2.1.1, as it depicts the current state of product development. Furthermore, reference should be made here to the latest edition of Pahl/Beitz (Bender and Gericke, 2021), a detailed standard work that presents the current state of methodical product development and mechatronic system

Four-cycle model

Pahl/Beitz

[1] In German: SPALTEN = **S**ituationsanalyse, **P**roblemeingrenzung, **A**lternative Lösungen, **L**ösungsauswahl, **T**ragweitenanalyse, **E**ntscheiden & Umsetzen und **N**achbereiten & Lernen.

development. Moreover, Ashby refers to it as *"The Bible – or perhaps more exactly the Old Testament – of the technical design field, developing formal methods in the rigorous German tradition"* (Feldhusen and Grote, 2013b, p. v).

Conclusion on classical methodologies

In a nutshell, it can be stated that VDI 2221, including its two sheets, generally represents the methodical procedure in product development based on iPeM in a very precise and up-to-date manner. VDI 2206 (V model), as the best-known procedure model in mechatronics development, is regarded as a standard work and is also up-to-date due to the extension to include cyber-physical systems.

2.1.3 Context description in product development

After the presentation of fundamental terms and the most established process models, the following section will take a closer look at the context of product development. Here, the *development type* and the *type of products (and services)* are the main aspects relevant for this work. Based on these, the need for agile working methods is derived. In addition, there are several other contextual factors, as Gericke et al. (2013) point out in their context factor framework for product development, which are not considered in this analysis.

Development type Although the various process models are generic due to their model character, the design of the work steps to be carried out in the actual implementation depends, according to Abeln, on the industry (automotive, aviation, shipping, power plant construction), the type of construction task (new construction, adjustment construction, variant construction), as well as the number of units produced (large series, medium series, small series, individual production). Although the industry is proprietary and the number of units depends on market demand, the nature of the development task is a determining factor that actively depends on the company itself. In order to uncover the innovation potential of companies, Albers et al. took a closer look at this vital aspect. Based on Schumpeter, who postulated an innovation to be the successful establishment of an invention on the market, the authors analyzed the correlation between the types of innovation and the development tasks. According to Henderson and Clark, there are four types of product innovation: incremental, architectural, modular, and radical innovation.[2] Although this classification of innovations is coherent from a retrospective perspective, market success cannot be predicted in advance (Albers et al., 2015).

For this reason, Albers et al. examined the respective shares of development types (new, variant, adjustment development) in greater depth in a study with 131 participants and broke them down per industrial branch. The results indicate that pure new developments are relatively rare, yet specific assemblies or parts often exhibit a high degree of novelty between product generations; hence, they developed the product generation development (PGE)[3] concept (Albers et al., 2015). This concept makes it possible to determine the share of innovation per component, which is more accurate than designating the entire product as a new, variant, or adjustment development.

[2] For more definitions, see Gausemeier et al. (2018).
[3] In German: PGE = Produktgenerationsentwicklung.

Type of products If the products are compared across generations, an apparent increase in complexity can be seen, both in terms of the number of individual components and their connections to each other. In the following section, the definition of a technical product or system, respectively, will be sharpened. The term *mechatronic* system was already coined in the 1960s and describes the interaction of mechanics, electronics, and informatics within a technical system. The basic structure of a mechatronic system is exemplified in Czichos (2019) and can be described as converting, transporting, and storing energy, material, and information flows with the help of sensors, information processing, and actuators. A detailed description of the interrelationships in mechatronic systems can be found in Czichos (2019) and a chronological overview of the development of mechatronic systems can be found in Isermann (2008).

According to Lindemann, there are currently three significant expansions of the classic mechatronic products: for one, product-service systems are characterized by enhancing the product performance by the integration of services. Secondly, the enrichment by cognitive elements towards a situation-oriented and learning behavior of the products (cognitive systems). Lastly, the ability of independent communication on the internet, paired with the triggering of actions in the product or the environment (cyber-physical systems).

Product-service systems

Product-service systems (PSS) are characterized by the fact that the actual business model is no longer just the sale of a good or the provision of services but the combination of both. Lindemann gives the example of a bike-sharing system. Compared to classic products, there are different and often more dependencies in PSS since, in addition to the conception and realization of service functions, other scenarios of product use must also be recorded, which in turn increases the complexity. The second and third circles of Gausemeier's four-cycle model describe this very clearly. For a more in-depth description of PSS, refer to (Lindemann, 2016, p. 871ff.).

Cognitive systems

Lindemann cites cognitive (or adaptive) systems as a second extension. While in a non-cognitive system the regulation is left to the human, in a cognitive system the autopilot in airplanes or assistance systems in cars carry out the actions. An example is Adaptive Cruise Control, a system trained by machine learning, which uses camera information on motorways to establish the distance to the vehicle in front according to user instructions and restores the original distance in the event of changes in the traffic due to other road users swerving in or out. This is particularly important for future advancements in technology such as autonomous driving.

Cyber-physical systems

The third expansion are cyber-physical systems (CPS), which have been defined by acatech[4] as *"a linking of real (physical) objects and processes with information-processing (virtual) objects and processes via open, partly global and interconnected information networks at any time"* (Czichos, 2019). As a concretization, Gausemeier et al. exemplify these as physical systems (such as a production machine or transportation vehicle) that feature inherent intelligence and communicate and cooperate via the internet. Intelligence in this context refers to the ability of the connected systems to cooperate in a goal-oriented manner (Gausemeier et al., 2018).

[4] National Academy of Science and Engineering. In German: acatech = Deutsche Akademie der Technikwisschenschaften.

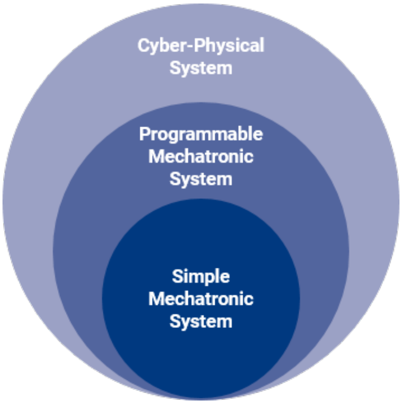


Figure 2.6
Development from mechatronic to cyber-physical systems according to Abramovici et al. (2016).

In figure 2.6, the evolution of mechatronic systems to programmable mechatronics systems and cyber-physical systems is displayed in an onion model, hence they are displayed as part of the model. According to Gausemeier et al., 15 billion products were connected to the internet in 2015, and this figure is predicted to double by 2020, which is a clear example of the increasing level of connectivity. Gräßler and Hentze illustrate this as well as the embedding of mechatronic systems in this "connected world" in figure 2.7 and elaborate this in the current version of the VDI guideline 2206 as follows: *"Cyber-physical systems (CPS) are created by expanding mechatronic systems with additional inputs and outputs that detect the environment and interact with it and coupling them to the IoT. Digital communication within a network occurs via wired or wireless infrastructures on a local or global level. Data transfer within the network is uni- or bidirectional. Both one-time discrete data exchanges and continuous real-time data transfers occur. [Figure 2.7] displays the shared value creation network (ecosystem) of a CPS consisting of a network of companies. The use of these services makes it possible to, e.g., adapt the system properties to the specific use of the user during the use phase within the product life"* (VDI 2206, 2020, p. 9).
A vivid example of providing and processing Big Data is the real-time display of currently available charging stations along the chosen route when, e.g., driving a Tesla automobile. In the future, this information will be provided to the autonomous vehicle acting as a cyber-physical system (and no longer exclusive to the user) so that the system can independently plan and adjust the route if necessary, in case all charging stations will be occupied at the predicted time of arrival at a specific charging location.
According to Eckert et al., experts from industry and science predict for the year 2040 that all products to be developed will be cyber-physical systems. A detailed description of CPS and its potentials can be found in VDI 2206 (2020). Lindemann summarizes that PSS, cognitive systems, and CPS cannot be sharply separated from each other and are essential building blocks of the future.

Need for agile product development *"The increase in technical changes, such as the intensive networking of technical systems, the digitalization of products and services, and the connection to the IoT, as well as organizational changes, such as increased complexity in communication between the disciplines involved, trends in project organization and the*

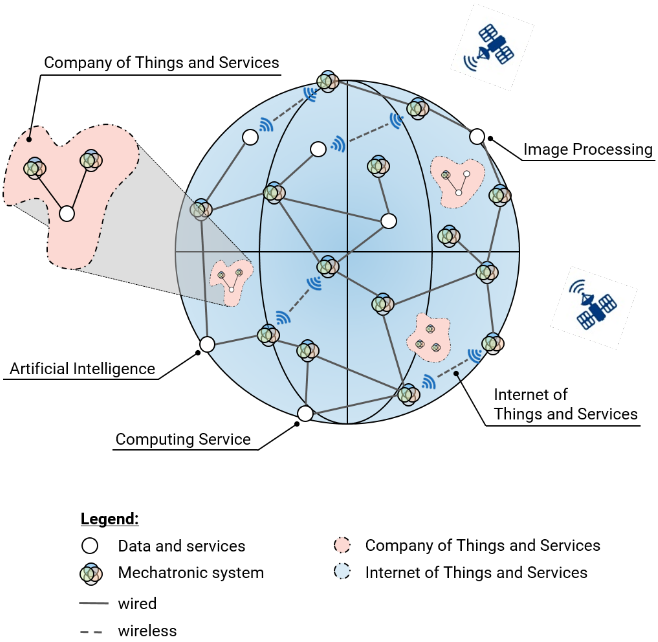


Figure 2.7 Ecosystem of CPMS according to Gräßler and Hentze (2020).

inclusion of strategic planning for the realization of digital business models", were reasons for the revision of the current version of VDI 2206 (2020). Feldhusen and Grote underline that the development of technical products undergoes constant adjustments due to changing boundary conditions. To classify these changes, de Weck et al. analyzed the sources of uncertainty-causing factors in product development and divided them into endogenous and exogenous factors. The clusters product, use, and corporate context are (rather) endogenous, whereas the market, political and cultural context are exogenous. Although these uncertainties are inherent in product development, the rate and speed of change are increasing rapidly. Due to an increase in variants, shorter innovation cycles, and a multitude of interfaces and thus increased product complexity, there is a need for new approaches in product development (Lindemann, 2016; Gausemeier et al., 2018).

The chapter aims to provide an overview of the current state of product development, its methodologies, and the products to be developed. The most recent references for the development of mechatronic products in the DACH region are VDI 2221 (2019a,b) and VDI 2206 (2020).

Summary of 2.1

2.2 Agility and agile product development

2.2.1 Understanding of agility

First of all, the term "agility" shall be defined. The work at hand builds on the definition of Böhmer et al. since she is a scholar in the field of agile product development compared to several other scholars who are active in the software or other non-mechatronic domains. Furthermore, this definition was acquired in the course of a systematic literature review conducted in the context of product development. See Böhmer et al. (2015) or Böhmer (2018) for a comprehensive overview of different meanings.

Definition of agility

"Agility is the capability to react, and adopt to expected and unexpected changes within a dynamic environment constantly and quickly; and to use those changes (if possible) as an advantage" (Böhmer et al., 2015, p. 4).

The first part of this sentence synonymously refers to flexibility; the second part after the semicolon distinguishes agility from sole flexibility, since an agile approach perceives the value of change as an advantage, whereas a flexible approach merely reacts to an occurring change (Schmidt et al., 2016). Agility needs to be understood as an attribute that can take different characteristics, such as agile companies (Dove, 1999), business agility (Kettunen and Laanti, 2008), or enterprise agility (Tsourveloudis and Valavanis, 2002) on a macro-level view. On a micro-level view, agility might be understood in a different way when shifting to individual domains such as agile manufacturing (Sharifi and Zhang, 2001) or agile supply chains (Christopher and Towill, 2001) up to agile products (Haberfellner and de Weck, 2005).

Agile product development

Agile product development, which is based on the idea of agility, is a relatively recent approach (Conboy, 2009). From the perspective of Information System Development and when comparing agile and lean approaches, Conboy has defined agile product development as a means *"to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment"* (Conboy, 2009, p. 340). In short, dealing with change is one of the foremost advantages of agile approaches. These changes originate from different sources; Krause and Gebhardt differentiate between internal sources, such as changing or incomplete requirements of stakeholders or (insufficient) tools or methods, and external sources such as technological innovations and evolving markets and therefore new competitors. Gericke et al. have developed a comprehensive list consisting of more than 200 factors, which exhibit an influence on the context of product development. These factors are classified based on five levels provided by Hales and Gooch and are concretized by three levels of granularity in a hierarchical manner.

These foreseeable and unforeseeable changes, which are the cause for uncertainty in the development process, result from a *"lack of knowledge about a problem at the point of time a decision on the solution of the problem is made"* (Lévárdy, 2006, p. 74). Uncertainty can either have positive implications in terms of opportunities, whereas negative ones are perceived as risks for the project (Lévárdy, 2006). Given the vast amount of possible uncertainty, agile development aims at utilizing these changes as an opportunity instead of focusing on preventing risks from happening as in classical risk management (Lévárdy, 2006; Ulrich and Eppinger, 2016; Schmidt, 2019).

Fundamental logic of agile development

As agile methods are adaptive and responsive to changing situations, they facilitate early feedback to evaluate often and, in line with this, guide development

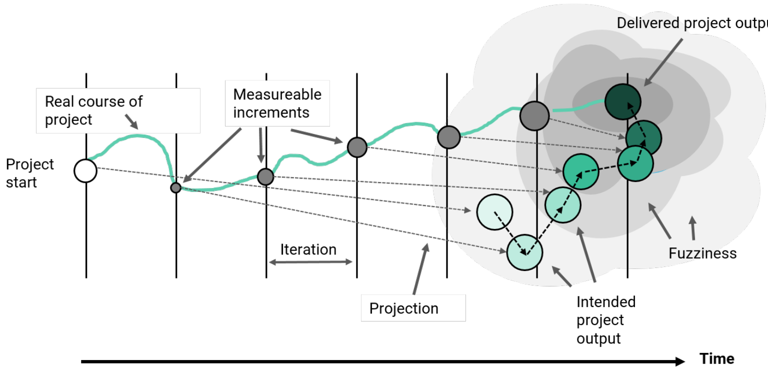


Figure 2.8
Fuzziness model according to Oestereich and Weiss (2008).

teams in terms of product quality and engineering efficiency (Douglass, 2016; Böhmer, 2018). In order to illustrate the fundamental logic of agile development, the fuzziness model by Oestereich and Weiss is chosen. Figure 2.8 displays how agile development deals with fuzziness (i.e., uncertainty) when either the project goal or the means to achieve a particular goal are unclear (or both) while satisfying the customer adequately.

An agile project is carried out in an iterative manner, represented by the space between two vertical lines in figure 2.8. The development project starts on the left and moves successively, in iterative steps, to the right until the project is finished. In the beginning, based on the initial information or requirements stated by the (potential) customer, the development team projects the intended goal (based on their understanding of the outcome) to a specific due date, which is visualized as a light green circle (aimed project output). This initial solution statement typically involves a high degree of uncertainty regarding the exact type of solution desired or the technology to be used. This uncertainty is represented by the larger grey areas representing the "fuzziness". Within every iteration, the team develops a measurable artifact (i.e., increment), representing one or more particular aspects of the final product. At the end of the iteration, the development team presents the increment to the customer and asks for feedback. By doing so, hypotheses regarding specific uncertainties (set at the beginning of the iteration) are tested. As a result, the team understands what the customers actually want versus what they initially stated. At the same time, the customers are aware of what the final solution might look like – and, based on these learnings, might change their mind due to certain aspects or insights. Therefore this short-cyclic procedure is actually a continuous, short-cyclic learning process for both parties resulting in a steady decrease of fuzziness from increment to increment and a steady increase in the refinement of the project goal over time (Oestereich and Weiss, 2008).

Similar models such as the OODA loop (Observe-Orient-Decide-Act) by Boyd or the AEZ[5] by Stelzmann are used to describe the fundamental logic of agile development as well. They all share that the procedure is not a linear sequence of repeating steps but rather continuous optimizations due to repeating feedback loops. All of these models are aiming at the early and fast exploitation and

[5] Agile development cycle. In German: AEZ = **a**giler **E**ntwicklungs**z**yklus.

response to changes. Therefore, agile development tries to hit a moving target (Schmidt, 2019). The development in small steps is an essential working principle of agile development, and the shorter the iterations, the more often a development team can pro- and reactively maneuver (Stelzmann, 2011; Schmidt, 2019).

2.2.2 Terminology and paradigm of agility

Terminology

Before diving into the historical evolution and paradigm of agile development, a few remarks regarding the terms are required. The terminology used in the further course of this work follows the Taxonomy for agile development by Atzberger et al. (figure 2.9, top) to ensure a uniform understanding. This taxonomy aims to agree on one terminology since classical and agile product development label the same elements (e.g., Scrum) with different terms. Agile practitioners call it an *agile method*, whereas it is referred to as a *methodology* in classical product development literature (Schmidt, 2019). From a structural point of view, the *St. Gallen management model* of the second generation by Bleicher is used as a classification scheme. It distinguishes between elements on the normative, strategic, and operative layer in a hierarchical manner, and classifies terms according to their characteristics with respect to structure, activity, and behavior on the horizontal axis.

St. Gallen management model in a nutshell

The model represents a systemic approach to explaining management tasks (Bleicher, 1991). The vertical direction reflects the time horizon for the effectiveness of decisions and is divided into the following three levels: normative, strategic, and operational. At the *normative* level, general corporate objectives are defined and superordinate values and behavioral norms are derived from them. These are anchored in the corporate philosophy. At the *strategic* level, the long-term business development strategy is described. For this purpose, the resulting procedures are defined based on performance potentials in order to implement the corporate principles. Operational management addresses short- and medium-term tasks in realization. At the *operative* level, the current concrete activities of the company are determined and controlled with regard to the set objectives. For an in-depth description of the model, see Paetzold (2020).

Since the St. Gallen management model is a scientific sound model that has been revised to the fourth generation, it serves as a solid foundation for the taxonomy. However, the second generation of the model is used due to its simplicity and clarity. The terms included in the taxonomy are scientifically grounded in product development literature to avoid ambiguity, which is explained in detail in Atzberger et al. (2020b).

Exemplary classification of Scrum

For the sake of clarification, *Scrum* and a few of its elements are inserted into the Taxonomy for agile development, as displayed in the bottom of figure 2.9. Agile development is placed onto the normative layer as the overarching development philosophy. Based on that, the so-called agile methods, which are actually methodologies, are located on the strategic layer. In the case of Scrum, the macro-level (several sprint loops) representation is located here as well, whereas the micro-level process model (single sprint loop with Daily Scrum mini-loop) representation is located on the operative level. In addition, a few elements of

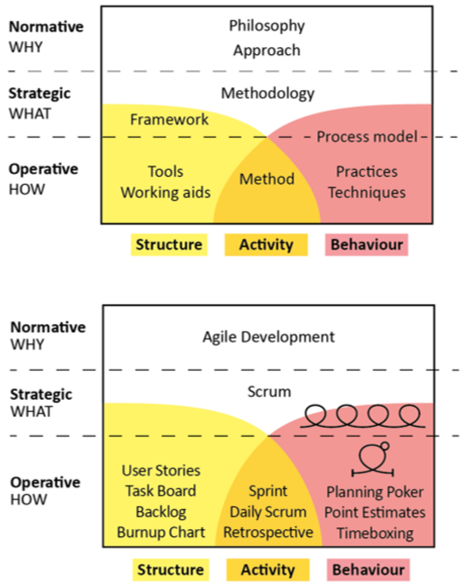


Figure 2.9
Outline of the Taxonomy for agile development (top) and exemplary classification of *Scrum* (bottom) according to Atzberger et al. (2020b).

the Scrum methodology are inserted on the operative layer to demonstrate how structural elements differentiate from behavioral ones. It is important to mention that some elements show characteristics of both ends, which is why these are placed in the *activities* section. Such operative elements are often referred to as *practices* in literature written by agile practitioners.

Paradigm

Agile process models have been developed as a response to prevailing impractical, heavyweight process models, characterized by long development cycles in the late 1980s and '90s (Beck, 1999; Highsmith, 2000). Even though classical process models have evolved over time and have been adapted to changing requirements accordingly, a new stream of methodologies has been developed by practitioners, especially from the software domain. *"Core to agile software development is the use of 'light-but-sufficient' rules of project behavior and the use of human- and communication-oriented rules"* that govern projects and embrace change (Cockburn, 2002, p. 8). In earlier times, the term *lightweight* was chosen to describe these approaches to highlight their dichotomy to the heavy-weight processes and indicate their quick responsiveness and high versatility (Cockburn, 1998). It is important to mention that the basic principles such as iterative and incremental design (Larman and Basili, 2003) and frequent working demonstrators (McCracken and Jackson, 1982) were not new even at that time; they date back to the 1950s and '60s. However, their recombination and application to the software domain were unique and thus groundbreaking. As the chronicle in chapter 2.2.3 shows, agile development has its roots in classical development, but it dissociates from it in the '90s by establishing a new paradigm (Schmidt, 2019).

Manifesto

The issue of the *Manifesto for Agile Software Development* can be seen as the cornerstone in several ways. For one, at that meeting back in 2001, seventeen experienced practitioners coined the term *agile development* for the first time, as an advancement of lightweight (Beck et al., 2001). Second, since all of these practitioners were software engineers who had been active in either developing or applying agile software methodologies, their aim was to discuss and grasp the foundation of this "new" way of developing code. As a result, the Manifesto consists of the established four values and twelve (axiomatic) principles that reflect what agile development characterizes in the authors' opinion (Highsmith, 2000). For an overview of the values and principles of the Manifesto, see Beck et al. (2001). As of today, 20 years after the issue of the Manifesto, no changes nor adaptations to the original values or principles have been made. The Manifesto, with its values and principles, is still considered valid, even when being applied in other domains. However, (and this has been pointed out in the introduction) the non-alignment has caused crudities regarding the initial purpose of agile development as well as its further use. For the sake of clarity, the values and the (axiomatic) principles of the Manifesto are of normative nature for the most part when inserting them into the Taxonomy of agile development. They are of guiding nature but, and this is crucial, they were developed in and for the context of software. Other contexts bear different restrictions, which will be addressed in chapter 2.3.1.

2.2.3 Chronicle of agile process models

In this chapter, a chronicle[6] of historically significant agile methodologies has been created. Every methodology included in this timeline is or has been used to some extent by practitioners and is mentioned by several sources (Komus and Kuberg, 2017; Schmidt et al., 2018b, 2019; VersionOne, 2019, 2020; Atzberger et al., 2020a). However, "mixed" approaches are not included in the chronicle. There is a broad range of combinations of existing approaches (more than 20 mixed combinations), as Heimicke et al. have shown, yet no statement can be made about the range of their use. *ScrumXP* and *Scrumban* are an exception here since these are prevalent combinations that have been used by a certain percentage of practitioners (VersionOne, 2020).

Outline

The chronicle containing agile process models/methodologies is presented in figure 2.10. It displays the first mentions of the respective methodologies as a red symbol and the first official scientific record as a blue symbol. Due to this, it has its beginning in the 1940s, since the essentials of both Kanban and Lean date back to this time. A green symbol indicates a significant further development of the respective methodology, such as shifting into another domain or an advancement of the methodology itself. The methodologies are sorted according to their first official publication and displayed consecutively. Scaling frameworks are also listed and highlighted by a dark blue background.

The chronicle is further divided into three periods of time, which is one result of the analysis that is being preponed for better comprehensibility. Period 1 ranges from the 1940s until 1990, being the *Pre-Software Period*. Period 2 ranges from 1991 until 2006, being the *Software Period*. Period 3 ranges from 2007 until today, being the *Scaling Period*. A short description of the methodologies follows hereafter, in chronological order and with respect to the periods.

[6] A chronicle is a qualitative research method of historical perspectives (Thomas, 2003). In comparison to a mere timeline, it provides additional information of the context and the given circumstances, which is provided by the subsequent analysis at the end of this chapter (Thomas, 2003).

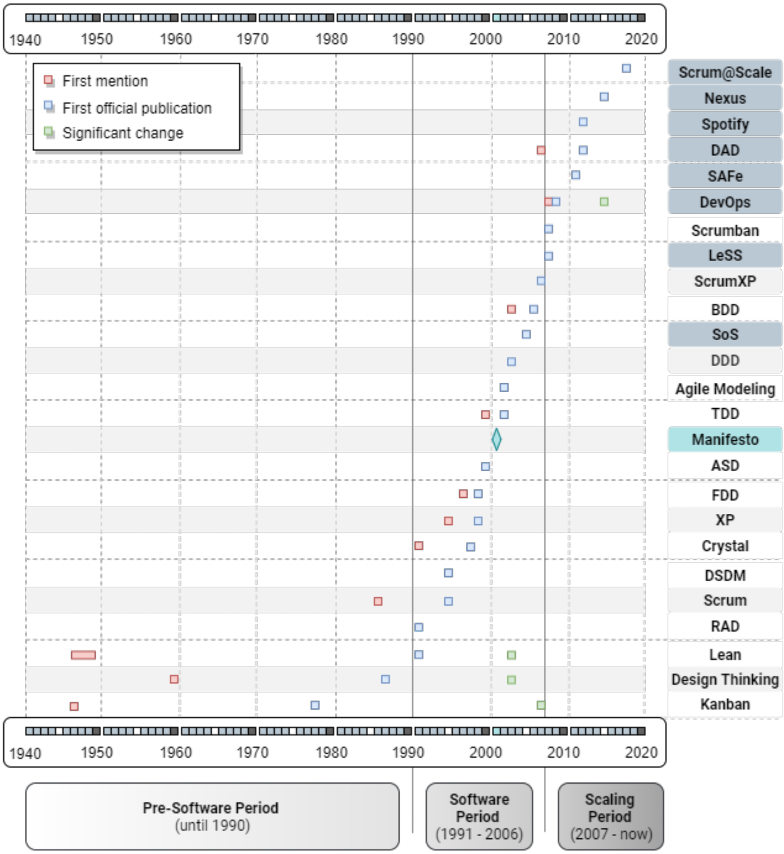


Figure 2.10
Chronicle of agile process models.

Pre-Software Period (1940s - 1990)

Chronicle of agile process models As illustrated in figure 2.10, three methodologies range back to more than half a century ago, namely Lean, Kanban, and Design Thinking. The core ideas behind both Lean and Kanban were developed in Japan, which marks the turning point for the industrial progress in the Japanese manufacturing industry as it is today. **Lean** focused on improving the system of product delivery by eliminating waste (everything not adding functionality to the final product is considered waste here). By contrast, **Kanban** aimed at optimizing the manufacturing process (flow) by regulating the supply of materials. Moving two decades ahead across the Pacific, **Design Thinking** has been come up with in the United States at Stanford (as a course in the field of mechanical engineering, ME310), intending to boost user understanding amongst designers (Arnold, 1959).

In the ’70s, the concept of Kanban spread to Europe and the United States, and it was published in 1978 for the first time under the term known as "Toyota Production System" (Ohno, 1988). Here, Lean concepts were also described since both methodologies were applied in the organization and production systems at Toyota. In the early ’90s, the principles behind Lean were published under the term "Lean management" for the first time, showing the benefits of just-in-time production at Toyota (compared to mass production at General Motors at that time) (Womack et al., 1991). In the late ’80s, the term "Design Thinking" was coined and published for the first time (Rowe, 1987). In the early ’90s, it was the primary procedure model of the well-known design company IDEO. In 1986, the term "Scrum" was used for the first time, describing the development success (yet also downsides) based on a few case studies at hardware development companies (Takeuchi and Nonaka, 1986). The first known publication of agile methodologies was not until the early ’90s.

Software Period (1991 - 2006)

In 1991, Rapid Application Development (**RAD**) was presented as a further development of the Spiral model by Boehm, which inherited concepts of working in an iterative (time-boxed) and incremental (rapid prototyping) manner (Martin, 1991). Thus, it can be considered one of the early agile methodologies for software development. Dynamic Systems Development Method (**DSDM**), which ought to be a project delivery framework specifically for RAD, was released by the DSDM consortium in 1994 (Stapleton, 1997). Since then, it has constantly been further developed, changing its name to DSDM Atern in 2007 and back to just DSDM in 2014 due to misunderstanding. Once only available for members, the DSDM handbook was made available publicly in 2014. Adaptive Software Development (**ASD**), which was published in 2000 for the first time, has its roots in RAD and focuses on continuous process adaptation by learning and adapting in the course of the project (Highsmith, 2000). Another methodology that has its roots in the early ’90s is the **Crystal** family. It incorporates different variants in the form of colors (clear, yellow, orange, etc.), which depend on the number of people involved and the criticality of the project (Cockburn, 1998).

Scrum had its first appearance in the software development industry in the early ’90s, with the first publication about the principles of Scrum in 1995. The official book on agile software development with Scrum was released in 2001 (Schwaber and Beedle, 2001). eXtreme Programming (**XP**) was developed and applied first in 1995 at the company Chrysler. The first publication on XP as a development methodology was published in 1999 (Beck, 1999). Since XP was not designed for large-size teams, the methodology Feature-Driven Development (**FDD**) was developed in 1997. It focuses on breaking down the overall model into several features, which create an added value and then successively working feature by feature. The first publication on FDD was released in 1999 (Coad et al., 1999).

Manifesto

As noted in the previous chapter, the Manifesto for Agile Software Development has been agreed upon and published in 2001 (Beck et al., 2001). It is marked as a turquoise diamond in the chronicle.

In 2000, Test-Driven Development (**TDD**) was developed based on XP since it was not focusing enough on the actual testing of the products, with the first book on the topic being released in 2002 (Beck, 2002). **Agile Modeling** has first been published in the same year and focuses on applying modeling techniques in an effective manner. It is intended to be tailored to fit other processes and support those by the techniques provided in the methodology (Ambler, 2002). In 2003, Domain-Driven Design (**DDD**) was developed and published. It deals with the suitability of the subject-matter knowledge of the domain it is designed for (Evans, 2003). Behavior Driven Development (**BDD**) was first created in 2003 as a further development of TDD. BDD focuses on the cooperation between quality management and business analysis in software development projects by capturing the intended requirements of the software (thus its behavior) using an ubiquitous language to run automated tests subsequently. The requirements are commonly phrased using the language of DDD, and it was published in 2006 for the first time (North, 2006).

As stated in the Manifesto, agile development claims to have its sweet spot (the most promising level of successful applicability, cf. chapter 2.4) in small, co-located teams. However, since the industry has experienced the benefits of working in an agile way, the founders of Scrum have come up with Scrum of Scrums (**SoS**), which can be considered an addition to regular Scrum, displaying how multiple teams can work in parallel. The first official publication on SoS was released in 2005, and it represents the first attempt to scale Scrum (Sutherland, 2005). In 2007, the first concept of combining two methodologies was presented and published, with **ScrumXP** being a blend of Scrum and XP. It incorporated the process structure of Scrum serving as a guideline for development sequence and the elements of XP focusing on the development activities (Kniberg et al., 2007).

Shifting back to **Lean**, its usage in software was published in 2003 under the term Lean Software Development, restating traditional lean principles and a set of tools on how to transfer them to the discipline of software development (Poppendieck and Poppendieck, 2003). In 2005, **Design Thinking** made a great leap forward by establishing the Hasso Plattner Institute of Design (known as d.school) at Stanford. The Hasso Plattner Institute School of Design Thinking in Potsdam, Germany, was founded only two years later. Moreover, the principles of **Kanban** were transferred to the software industry as well, known as IT-Kanban. The initial concept is enriched with Lean Development principles, risk management, and the Theory of Constraints and was presented in 2007 for the first time (Anderson, 2010).

Scaling Period (2007 - today)

Whereas in the Software Period several different methodologies have been developed, the following period shows different characteristics: After having experienced the benefits on the team level, the methodologies developed in this period are either focusing on the collaboration across different teams in order to scale agility inside the company or combining existing methodologies. In 2008, Large Scale Scrum (**LeSS**) was developed to run several Scrum teams in parallel to achieve a common goal. Here, multiple teams are collaborating by having only one Product Owner and a common Backlog overall, whereas in SoS, each team has its own Product Owner and Backlog (Larman and Vodde, 2008). LeSS has also further developed by having two versions, LeSS Basic (for up to eight teams) and LeSS Huge (more than eight teams). In the case of LeSS Huge, the overall Backlog is subdivided into Area Product Backlogs with an Area Product Owner each. **Scrumban**, as a combination of Scrum, Kanban, and Lean, has been published for the first time in 2008 as well. The idea was to combine practices and principles from each methodology and synthesize them into one methodology – a "best of breed" methodology (Ladas, 2008). The following year, **DevOps** was introduced, which aims at spreading beyond the development and combining both *Development and Operations*. Due to a closer consolidation of these two areas, a higher deployment rate and a decreasing failure rate is intended to ensue (Debois, 2009). Further advancements also include additional fields, such as BizDevOps (+ business) or DevSecOps (+security).

In 2011, the Scaled Agile Framework (**SAFe**) was introduced. It consists of three levels (portfolio, program, team level) with specific roles at each level and separates the work into value streams. There are one or more Release Trains within a value stream, with five to fifteen Scrum teams each. Due to its manifold character, its suitability ranges up to large enterprises (Leffingwell, 2011). While it had been mentioned already in 2007, the first official report about the Disciplined Agile Delivery (**DAD**) framework has been published in 2012. It considers itself a hybrid agile toolkit, which adopts elements from different methodologies (Ambler and Lines, 2012). In the further advancement, additional improvements have been made and enlarged its scope from DAD to Disciplined DevOps, Disciplined Agile IT to the most recent overarching area, which is referred to as Disciplined Agile Enterprise (DAE) today (Disciplined Agile Consortium, 2017). In 2012, another agile methodology developed by a prominent music streaming company was presented, the so-called **Spotify** model.[7] Having started by applying Scrum, the company changed their process model so that their teams are divided into what is referred to as Squads, Guilds, and Tribes. Their model works for the large-scale product development department and is guided by both autonomy and alignment. The initiators also stated that what they were presenting back in 2012 was just a snapshot of their company's current organizational and processual state and is subject to change when needed (Kniberg and Ivarsson, 2012).

In 2015, one of the founders of Scrum developed a framework called **Nexus**, which relies on Scrum. They state that it is intended for up to nine Scrum teams working on a single Product Backlog. Even though it has some additions to the original Scrum Guide, Nexus and its process model are close to Scrum, focusing on the Integrated Increment at the end of a sprint (Schwaber and Scrum.org, 2015). In 2018, the second founder of Scrum had published **Scrum@Scale**, referred to as a meta-level framework and relies on "basic Scrum". It consists of a Product Owner and a Scrum Master Cycle. Every cycle needs to answer a specific set of questions in order to be able to scale "regular Scrum". Due to the absence of strict guidelines, it enables establishing a tailored "agile scaled framework" (Sutherland and Scrum Inc., 2018).

Origin in the hardware

Remarks on the chronicle The concept of agility in the software industry, as many people are referring to, has its beginning in the early '90s. Yet, it has its roots outside the software industry, as the Pre-Software Period indicates – Kanban and Lean have their origin in the manufacturing industry, and Scrum and Design Thinking in physical product development. The early concept of agility did not call itself agility, nor was it as comprehensive and detailed as it is today. However, it is crucial to understand that concepts from other fields have been taken and they were adapted to the software discipline – which did result in a great success, as we know today, according to recent studies on the field (Komus and Kuberg, 2017; VersionOne, 2019, 2020). The linkages to its early roots become evident, especially when taking a closer look at the scaling period.

[7] For a detailed description, see chapter 7.4.

Table 2.1
Overview of the origins of the methodologies and scaling frameworks.

Year	Methodology	Origin
1995	DSDM	RAD
2000	ASD	RAD
2002	Agile Modeling	XP
2003	DDD	XP
2005	SoS	*Scrum*
2006	BDD	TDD
2007	ScrumXP	*Scrum*, XP
2008	LeSS	*Scrum*
2008	Scrumban	*Scrum*, Lean, Kanban
2009	DevOps	*Scrum*, XP, Lean, Kanban
2011	SAFe	*Scrum*, Lean
2012	DAD	*Scrum*, XP, Lean, Kanban, AM
2015	Nexus	*Scrum*
2018	Scrum@Scale	*Scrum*

Adaptations and advancements in the software

The founding fathers of the early agile software methodologies have recognized the impracticability of the prevailing "heavyweight process models" and understood the advantages of using "lightweight methodologies", which they have developed and then published in the following years. In this respect, two streams have emerged in the Software Period: A *subject-related* and a *process-related* stream. Methodologies following the subject-related stream (e.g., XP) are dealing with the actual handling of the development task, i.e., how the development of the product shall be carried out by applying certain practices such as, e.g., Pair Programming or Refactoring. The most representative methodology in the second, process-related stream is Scrum since the product's actual design is not supported in any way, but the coordination of the people involved in the development process. Several methodologies incorporate aspects of both streams, but they show an inclination to one stream. Table 1 summarizes all methodologies, which have emerged from an earlier methodology. The first further developments, namely DSDM and ASD, rely on RAD. After the turn of the millennium, Agile Modeling and DDD were further developments of XP, aiming at improving certain design aspects within the design process. With the beginning of the Scaling Period in approximately 2007, Scrum has been the foundation for the methodologies and scaling frameworks ever since.

At the beginning of the Scaling Period, the first so-called "hybrids" originated, meaning the combination of two or more methodologies into a new or modified version to attain a "best of breed" of the different process models. Here, ScrumXP was created by merging Scrum and XP, a blend of two diverse streams combining both subject-related and process-coordinating aspects into one methodology. Scrumban, on the other hand, can be considered a more flexible, and therefore rather "lightweight version of Scrum" by incorporating ideas from Lean and Kanban. Besides these two hybrids, several other mixtures have been proposed and were published over the years, e.g., Xanpan, eXSRUP, SCR-FDD, or a combination of DSDM, Scrum, and XP or RUP and Scrum, only to name a few. However,

none of these hybrids have reached a broad acceptance in industrial practice, which is why they have not been considered in the chronicle. While hybrids aim at complementing each other, most hybrids contain (elements of) Scrum according to Heimicke et al. (2020).
Shifting to the scaling frameworks, each of the frameworks mentioned above relies on Scrum or has taken specific elements out of it, as displayed in table 2.1. Only the Spotify model cannot be considered a "pure" further development of Scrum, even though it was developed based on the learnings by the usage of Scrum, since Spotify started with Scrum in the early beginning. The further development of the methodologies in the chronicle shows that since every scaling framework relies on Scrum, it can be summarized that the process-coordinating character is a critical success factor in the Scaling Period.

Summary of 2.2

In the following chapter, the application of agile methodologies in mechatronics development is investigated. An overview of the current usage of agile mechatronics development is given in chapter 2.3.2. For a general overview regarding agile process models, see VersionOne (2020).

2.3 Agile development in mechatronics

Classical (plan-driven) approaches[8] have been elaborated in chapter 2.1, whereas agility and agile software development have been explained thoroughly in chapter 2.2. In this chapter, the differences in the development of a mechatronic product compared to a software product are examined in more detail. In other words, agile development is investigated outside the software domain, in the context of mechatronic product development. This clarification appears to be redundant, yet it is vital to understand how the development of software differs from mechanical product development.

2.3.1 Disassociation to agile software development

Context of mechatronics

The evolution and advancement of mechatronics has already been described in chapter 2.1.3. Another rather plain representation of what mechatronics consists of is presented in figure 2.11. This illustration has been added to the latest edition of VDI 2206, as it includes other disciplines besides mechanics, electronics, and software. The adaptation emphasizes the increased need for interdisciplinary collaboration in design, representing a new era product-wise – the shift towards cyber-physical mechatronic systems (Guaragni et al., 2016; Paetzold, 2017). Figure 2.11 displays the interconnected nature of a mechatronic product and, therefore, its scope compared to a pure software product in terms of product complexity.

Remark

The terms *CPMS*, *CPS*, *hardware*, and *mechatronic products / systems* are being used interchangeably in this work. Even though there are differences between those

[8] The term *plan-driven* is specified as follows: As agile approaches were once called lightweight approaches, as counterpart to heavyweight process models, plan-driven is used as a synonym here to describe the once heavyweight / traditional / classical approaches.

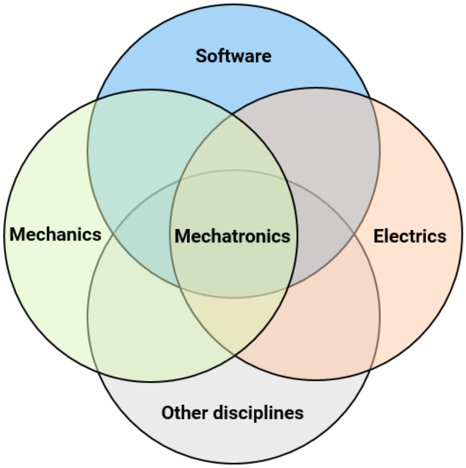


Figure 2.11
Structure of a mechatronic system according to VDI 2206 (2020).

terms as well as between a product and a system, this thesis focuses on characterizing the distinctions of the tangible nature (physical share) of a mechatronic product compared to the virtual nature of a software product. Hence, for the sake of readability, the terms **mechatronics** and **mechatronic products** is used as umbrella term from this point on.

The denotation "physicality" might seem odd from a mechanical engineering point of view; however, this differentiation is logical and consistent when considering a software engineer's viewpoint. In line with this, when aiming at educating non-software engineers in applying agile development in their development processes, the physicality (tangible share) of a mechatronic product bears challenges due to the fact that agile principles and practices cannot be transferred to the context of hardware development without further ado (Ronkainen and Abrahamsson, 2003; Conforto et al., 2014). For them to be applicable in the context of mechatronics, adaptations are mandatory due to the distinctions of software and mechatronic products (Conforto et al., 2014; Schrof et al., 2018).

The differences regarding the effort of the main activities in creating either type of product are visualized in figure 2.12 (Socha and Skip, 2006). The font size represents the magnitude of effort needed to fulfill a task. Throughout the entire lifecycle of a software product, the design phase takes up most of the time, as the product needs to be written in code. The following building and distribution can be neglected resource-wise. When shifting to the development of a physical product, the building of the actual product takes up the highest amount of resources compared to its design. The subsequent distribution and operability are resource-intense throughout the lifecycle of the product as well. As agile development stresses the incremental and continuous generation of prototypes, the high resource effort in terms of building these is impedimental. In short, *"physicality heavily constraints the development of physical products"* (Schmidt, 2019, p. 28). As a result, these restrictions are referred to as constraints of physicality when applying agile development to mechatronic products (compared to software products).

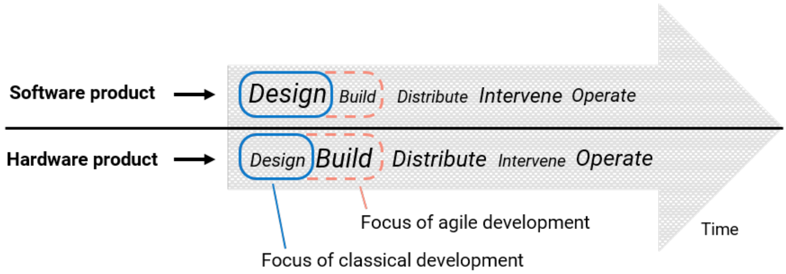


Figure 2.12
Lifecycle differences regarding time and resource effort between the creation of software and hardware products; based on Socha and Skip (2006), adapted from Schmidt (2019).

Problem areas

The term "Constraints of Physicality (CoP)" was first coined by Ovesen when investigating the application of Scrum in seven Danish companies in 2012. Overall, he identified five general conditions, which represent different clusters of challenges. The constraints of physicality are one of these problem clusters, which can be summarized as the *"shift from the freedom of virtuality in software development to constraints of physicality in integrated product development"* (Ovesen, 2012, p. 151). *Paradigm perplexity* as another cluster relates to the transformation of a radically new development paradigm (methodical framework) into a traditional development environment. The cluster *designer's dissent* refers to the different perceptions of individuals in approaching new methodologies by either adopting or reacting with uneasiness towards these. The cluster *team distribution dilemma* characterizes the differences in deploying and organizing development teams. As the application of any new paradigm requires training, *education and maturation* presents the last general condition. See Ovesen (2012) for more information regarding these clusters of challenges. Schmidt et al. have identified additional challenges, such as *shippability issues* and the *management of external dependencies*, such as development partners and suppliers. Atzberger and Paetzold have examined how the challenges once identified by Ovesen have developed over the years and which ones are still of importance compared to seven years ago. They found out that the constraints of physicality are still considered to be a significant hindrance by practitioners. The second-largest challenge cluster is considered to be the *mindset* of the people, referring to the understanding of what agility is capable of handling in terms of education, training, and transformation. The *team distribution* issue is still up to date and seen as challenging. The fourth challenge cluster represents *scaling* agile development within the companies, which is a new issue.

In order to address these challenges, Stelzmann states that three strategies exist to deal with these. Either by ↪ using existing methods and principles in an unconventional way or by ↪ trying to fulfill the prerequisites so that the methods can be applied just as in the software industry or by ↪ developing new practices and principles that fit the context of agile product development (Stelzmann, 2011). As the reusability of existing methods from software development in the context of mechatronics is somewhat limited (Schrof et al., 2018), and changing the prerequisites is almost impossible due to physicality reasons, Schmidt has chosen the third option by contributing ten principles and three practices specifically designed for the context of mechatronics. Nevertheless, open threads remain in the context of mechatronics, as shown by the latest cluster of challenges.

Similar research fields

For the sake of completeness, other approaches besides agile development, which are used in mechatronics development, are described in this paragraph. Schmidt has identified three research fields: For one, **flexible product development** by Smith focuses on responding to change in a flexible manner, i.e., pursuing several solution approaches by following a Scrum-like procedure with additions of set-based design[9], and thus delaying decisions to the latest possible point in time. By applying several practices from agile software development that aim to determine the actual customer value such as user stories or iterative development cycles with rapid customer feedback, the pursuit of different approaches shall increase flexibility. Furthermore, **adaptive product development** aims at adapting to the changing state of a development project instead of *"dictating a specific project schedule a priori"* (Lévárdy and Browning, 2009, p. 601). Therefore, by modeling a product development as a complex adaptive system, many adaptive cases can be simulated, and insights regarding the highest-value process emerge (Lévárdy, 2006; Lévárdy and Browning, 2009). **Lean product development** or lean innovation is based on lean production principles and has its origins in the first half of the 20th century, as displayed in the chronicle in figure 2.10. The Toyota Production System, which is based on lean principles, focuses on value creation by orientation on customer needs, waste reduction, set-based concurrent engineering, continuous improvement, and a high degree of integration in a lean value chain (Link, 2014). It can be considered a holistic development approach as it goes beyond applying process models to form a suitable development mindset (Schmidt, 2019).

2.3.2 Methodologies used in agile development of mechatronics

Agile product development is a holistic approach that covers the customer, project organization, team members, processes, and the product itself (Stelzmann, 2011). In line with this, it covers both product development (operative design activities) and project management (administrative organization activities) according to Schmidt et al. (2018b). An overview of the current fields of knowledge has been described in detail by Schmidt and is visualized in figure 2.13. In addition, a brief summary of the main topics and further readings is presented in table 2.2.

Fields of knowledge

The fields of knowledge are subdivided into three perspectives. From a *technical-physical perspective* (Figure 2.13, blue), the actual design of the product to be developed as well as the (product-related) change are current topics of investigation. Shifting to the *technical-management perspective* (red), the maturation and education of people on the one hand and the organization of both the project and team members on the other hand are relevant research areas. The *system-theoretical perspective* (green) investigates topics that are going beyond the original focus of agility to distributed development teams, application to large-scale projects, and the necessary prerequisites for agility of the ecosystem if applied outside the so-called sweet spot for agile development (cf. chapter 2.4).

[9] In contrast to *point-based design*, which focuses on one design option, *set-based design* pursues more than one and thus, keeps multiple design options open as long as possible. For more information, see Schmidt (2019).

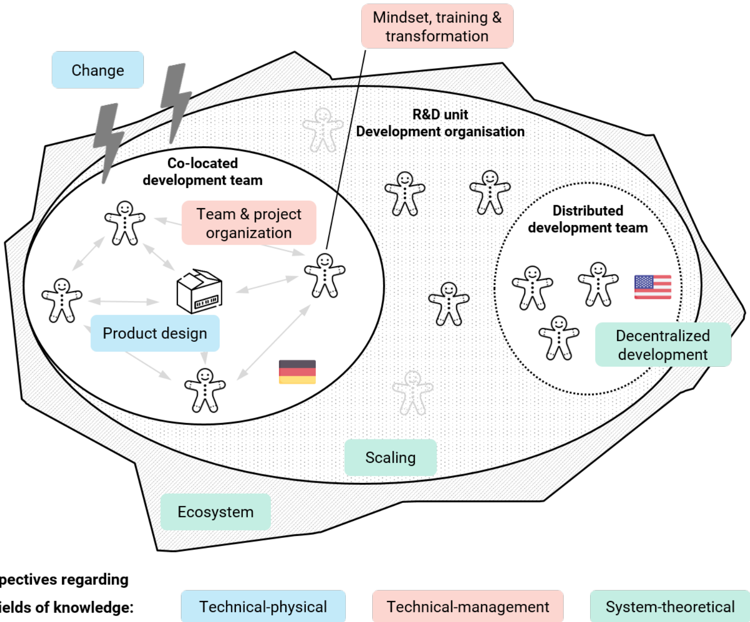


Figure 2.13
Fields of knowledge in agile development adapted from Schmidt (2019).

The work at hand focuses on the knowledge field *mindset, training & transformation* since the perception of agility in the context of mechatronics is being investigated. In order to remain focused on answering the research question, all other fields of knowledge, as displayed in figure 2.13, are excluded in the following chapters.

Current usage of agile methodologies

According to renowned current surveys on the application of agile methodologies and scaling frameworks in mechatronics development (Komus and Kuberg, 2017; Schmidt et al., 2018b, 2019; Atzberger et al., 2020a), it can be summarized that regarding "standalone" agile methodologies, **Scrum**, **Kanban**, and **Design Thinking** are by far the most used methodologies today. In addition to these, the hybrids ScrumXP and Scrumban are among the most used methodologies as well. Whereas Design Thinking is primarily used in the early stages of the product development process, Kanban is popular due to its simple and effective continuity throughout the entire development process (Böhmer et al., 2015). Scrum, however, suits many "traditional" companies due to its rather prescriptive rules and roles. Even though the work coordination in interdisciplinary teams and daily collaboration (of Scrum) is referred to as relatively new or uncommon and differs substantially from their status quo, the rules and roles give practitioners a certain routine and thus a sense of identity they can cling to. Due to these characteristics, Scrum was able to gain the broad acceptance it has today. For further discussion on the application of agile methodologies, see chapter 7.4.

Nevertheless, it is important to mention that in the early 2000s, XP used to be the most popular methodology among all agile process models. The reason for this were the distinct practices of XP specifically designed to address the subject-related issues in developing software. However, this methodology is not capable of handling and coordinating multiple teams or a large number of develop-

Table 2.2
Overview of the fields of knowledge based on Schmidt (2019).

Knowledge field	Description	Literature
Technical-physical perspective		
Change	Investigation of the sources and causes of changes, the need for agility and appropriate cases of application.	Thomke and Reinertsen (1998); Lévárdy (2006); Kruchten (2011); Stelzmann (2012); Bennett and Lemoine (2014)
Product design	Challenges and solution approaches in the design of products and their corresponding architectures.	Haberfellner and de Weck (2005); Fricke and Schulz (2005); Erickson et al. (2005); Ovesen (2012); Albers et al. (2017); Schuh et al. (2018); Schmidt (2019)
Technical-management perspective		
Mindset, training & transformation	Education and consolidation of the perception of agility to change the way of thinking and therefore enable a meaningful transformation process.	Conboy et al. (2011); Lalsing (2012); Gandomani et al. (2014); Dikert et al. (2016)
Team & project organization	Organization of team personnel as well as appropriate method application for a systematic elaboration of the product.	McMahon (2006); Stelzmann (2011); Ovesen (2012); Klein (2015); Schmidt (2019)
System-theoretical perspective		
Ecosystem	Investigation of prerequisites of a suitable ecosystem which suits the needs of agile projects.	Kruchten (2011); Böhmer (2018); Cooper and Sommer (2018)
Scaling	Challenges and solution approaches of scaling agility beyond the application of dedicated agile methodologies.	Ambler (2009); Alqudah and Razali (2016)
Decentralized development	Global applications of agile ways of working require adaptations regarding the methods and tools for decentralized collaboration and coordination.	Holmström et al. (2006); Hossain et al. (2009); Khan et al. (2011)

ers. This is why Scrum and the scaling frameworks based on this methodology started to get more attention since Scrum's focus is on the coordination of development personnel.

According to the surveys mentioned earlier, the current level of adoption of scaling frameworks states that **SAFe**, **SoS**, **DAD** and **DevOps** are the ones currently used most in practice. As shown in table 2.1, they all rely on or have developed based on Scrum, however not solely. The outline of Scrum is scalable, but it lacks essential aspects which are necessary when rolling out agility across the borders of R&D. This is why aspects of additional methodologies are incorporated into scaling frameworks (except for SoS, which is used to scale Scrum inside the R&D and can thus be interpreted as an intermediate step to an agile transformation). For an agile transformation to succeed, other principles are necessary to specify the coordination between departments (Schmidt, 2019). DevOps aims at aligning the development and operations in order to enhance the collaboration between these departments.

In contrast, DAD provides a broad toolkit from which the strategy department can pick which suits their company's needs best, whereas SAFe provides an overall outline divided into four layers on how to achieve and execute a rollout inside the company. The prescriptive character coupled with its clear and transparent outline (and its long-lasting availability) appear to be the critical success factors for it to be the most popular scaling framework overall. Important to specify is that the scaling frameworks mentioned above have in common that they incorporate additional principles (compared to standalone methodologies) in order to be able to scale agility across multiple teams and departments. As the focus of work is shifting beyond the barriers of single-team work, adaptations on the strategic layer (cf. figure 1.1) appear to be necessary compared to the operative process support by "standalone" agile methodologies.

Apart from the methodologies described above, a shift is recognizable in the latest surveys: A new category, own process models, placed fourth (39 % of 117 participants in total) in the study of Atzberger et al. (2020a) indicating that companies are developing their own, customized versions of agile process models that fit their companies' needs best. This shift is discussed in more detail in chapter 4.1. Given this trend, companies are looking for ways and means to apply agile development in physical product development and recognize that a sole "copying" of agile methodologies from software development without far-reaching adaptations is insufficient (see chapter 4.2).

Approaches and methodologies for agile mechatronics development In this section, approaches and methodologies specifically designed for (or adapted to) the context of agile development in mechatronics are presented. Following the classification of Wynn and Clarkson (2018), they are distinguished by their model type (macro-, meso-, micro-level). First, pure approaches and methodologies on a macro-level are displayed before diving into meso-level models. The third category comprises mixed or so-called hybrid approaches that specifically call for a combination of agile and classical (Stage-Gate) process models. In these paragraphs, the approaches and process models are presented in chronological order.

Cultural imprint

Moreover, all of the approaches and methodologies discussed here were designed within or aligned to the cultural boundary of the DACH region. As the work at hand focuses on influencing the mindset of people, cultural prerequisites have to be taken into account. Therefore, this constraint regarding the cultural imprint is set here and is valid from this point forward.

Macro-level

Pure approaches & methodologies The *Methodology for agile engineering* by Klein aims at providing guidance on how to transfer agile practices to the field of mechatronics. Depending on ten specific contextual factors, the individual agility level for the project at hand can be determined. For each agility level, suitable agile techniques are suggested. The *Agile Product Development* approach by Schuh et al. consists of five central paradigms that aim to support companies in a successful transformation from plan-oriented to highly flexible product development processes. These five paradigms identify central aspects that are important for developing in an agile manner, yet they are rather of strategic nature aiming to address future ramifications. Kantelberg has developed the *Methodology for designing agile development processes of technical products*, which aligns classical product development and agile software development using the ZHO model.[10] It provides a holistic overview of both the macro- and meso-level design of an agile product development process. The *Agile Systems Design* approach by Albers et al. accounts for the individuality of development projects by organizing appropriate fields of action in an agile manner wherever meaningful. It is characterized by nine basic principles that support teams in the development of mechatronic systems. These serve as guidelines for aligning activities and identifying suitable practices that support developers in the product development process.

In 2009, Feldhusen et al. have developed their concept of *agile methods to design for customer* (in analogy to "Design for X"). Their focus is on determining the actual customer need by predominantly creating virtual prototypes. They claim that there is an optimum level of agility, which ranges between predictive (planning costs) and agile (communication costs). Identifying this optimum is the main hurdle that has to be evaluated based on the respective project. The agile framework (*TAF*) has been developed by Hostettler et al. at the Technical University of Munich. Its micro-logic is based on the PDCA cycle ensuring the frequent generation of prototypes, whereas the meso-level logic focuses on addressing the topics of desirability, feasibility, and viability. It has been applied multiple times in both student and company projects (Böhmer, 2018). Schmidt has developed a *framework for agile development in mechatronics* that addresses the contextual differences between physical and virtual (software) products. Therefore, his framework consists of ten principles and three practices that facilitate agile development in mechatronic product development. Meso-level

Hybrid approaches In the following section, combinations of agile and plan-driven approaches are listed, which are referred to as hybrid approaches in this work.

In their work *Combined use of agile and stage-gate models in new product development*, Ahmed-Kristensen and Daalhuizen have investigated the applicability of both streamlines based on a qualitative assessment of four companies. They do not provide a hybrid process model, but they have identified relevant influencing factors which need to be considered, such as the degree of innovativeness of a product, the level of uncertainty, and product complexity. Having investigated the combination of classical approaches mixed with integrated product development (IPD) before, Sommer et al. have shifted their focus to combining agile and State-Gate approaches. In their work *Improved product development perfor-*

[10] Goal system, action system, object system. In German: ZHO = **Z**ielsystem, **H**andlungssystem, **O**bjektsystem.

mance through Agile/Stage-Gate Hybrids, they have developed the "Industrial Scrum Framework" that details into three hierarchical levels, with Stage-Gate models for strategic management at the top level and Scrum iterations to structure execution on the operative level.
Based on prior work, Conforto and Amaral have developed the *hybrid management framework IVPM2* (Iterative and Visual Project Management Model), which links elements of agile project management with Stage-Gate phases. It is a seven-step process that concretizes in three levels, making it a granular and thus practical process model. In 2016, the first edition of the *Agile-Stage-Gate Hybrid Model* was developed by Cooper. It is summarized as *"Stage-Gate [being] a macroplanning process and Agile [being] a microplanning project management methodology"* (Cooper, 2016, p. 22). The *"hybrid model balances the benefits and challenges of the two different approaches"* and is intended to be suitable for physical product development as well (Cooper, 2016, p. 26). Its application in additional domains is investigated based on six case studies in 2018 by the same authors (Cooper and Sommer, 2018). In 2020, its usage was elaborated on the portfolio level, as its adoption *"requires senior management to adopt a new mindset and use different metrics to gauge the health and progress of the portfolio"* (Cooper and Sommer, 2020, p. 32).
The methodologies and approaches mentioned in this chapter are manifold regarding their scope, dimension, and type of agility. In the following section, they are analyzed and compared regarding their explicit mention of normative elements.

2.3.3 Normative elements in agile development of mechatronics

All of the aforementioned approaches and methodologies have been developed for and within the context of agile mechatronics development. As this work investigates how to "adopt a new mindset" and sharpen the understanding of what characterizes agility in the context of mechatronics development, the approaches and methodologies are analyzed regarding their *normative characteristics*. This property is used to describe elements that are located on the normative layer of the St. Gallen classification scheme, which has been elucidated in chapter 2.2.2 when describing the setup of the Taxonomy for agile development, as displayed in figure 2.9.

Prelimiary defintion of terms

According to the definition of Epping, a **value** can be described as an abstract goal associated with a benefit. It should incorporate a fundamental character and make the procedure more authentic so that the user identifies with it (Epping, 2011, p. 14). As the Manifesto for Agile Software Development consists of four values and twelve axiomatic principles, the values represent convictions and frame what is considered desirable or crucial, and the **principles** concretize them by suggesting guiding rules or working mechanisms (Schmidt, 2019, p. 22). Therefore, values are superior to axiomatic principles, yet they are both located on the normative layer.
The original model of the Taxonomy for agile development is enriched by a value classification scheme (Atzberger et al., 2020b). When referring to figure 2.14, values such as *collaboration* and *empowerment* are placed on the normative layer. As

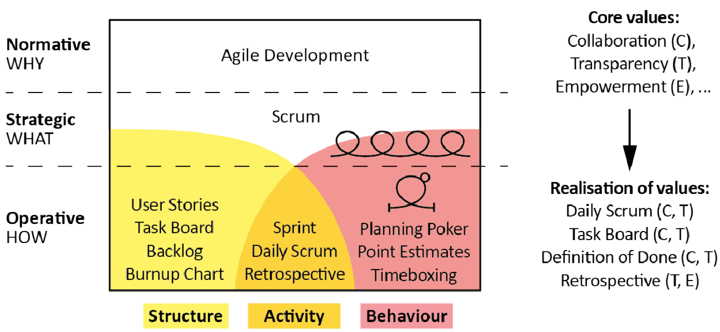


Figure 2.14 Taxonomy for agile development augmented by values according to Atzberger et al. (2020b).

the concretization of a value, a principle is, e.g., the co-located, interdisciplinary, and daily collaborative work routine.[11] Their subsequent translation into practice as working aids (e.g., User Stories) or methods (e.g., Retrospectives) is displayed on the operative layer.

⚠ Remark

Fuzziness and ambiguity regarding certain terminologies are a major issue in agile product development due to several reasons, which will be explained in detail in chapter 4.1. Therefore, the precise differentiation between values, principles and objectives is given in chapter 3.2 when elucidating the Design for Values approach. For now, the definition by Epping (2011) is sufficient.

The methodologies and approaches of the previous section are compared in terms of scope (macro, meso, micro), type (agile, hybrid), and value consideration (●●● = none, ●●● = partly, ●●● = profound) and are listed in table 2.3. All of these methodologies and approaches fulfill the preconditions of having been developed for the context of mechatronics, agile product development, and being suitable for an application in the German-speaking (DACH) region, which is why these categories are not explicitly listed.

Relevant publications

The publications by *Klein (2015)*, *Kantelberg (2018)*, and *Albers et al. (2019b)* are investigated in more detail, as these methodologies and approaches profoundly deal with values in their works or at least to some extent. They are elaborated in a more detailed manner, and it is shown how values are used in these methodologies.

Klein (2015)

The **methodology for agile engineering in machine and plant engineering** is one of the first publications which aims at transferring the working mechanism of agile software development to the domain of machine and plant engineering. The methodology follows a seven-step process, which can be summarized in a condensed way as follows: By acquiring a specific use case, initial context factors are determined, and the use case is classified by identifying the appropriate agility class (the author has developed these agility classes). For each agility class, a predefined set of agile practices is provided. Based on a given reference process, the target process can be adapted given recommend agile practices in different phases within the process. In the end, measures are derived and implemented. In his conception, Klein builds on the causal chain "value – element – technique", meaning that a value (e.g., transparency) is realized by an element describing a particular activity (i.e., dealing with transparent infor-

[11] See principles 4 and 6 of the Manifesto (Beck et al., 2001).

Table 2.3
Comparison of methodologies and approaches in agile mechatronic development.

Methodology	Scope	Type	Value consideration
Methodologies in the context of agile product development (DACH region)			
Schuh et al. (2017)	Macro	Agile	●○○
Feldhusen et al. (2009)	Meso/Micro	Agile	●○○
Hostettler et al. (2017)	Meso/Micro	Agile	●○○
Schmidt (2019)	Meso/Micro	Agile	●○○
Ahmed-Kristensen and Daalhuizen (2015)	Meso	Hybrid	●○○
Sommer et al. (2015); Cooper (2016); Cooper and Sommer (2018, 2020)	Macro/Meso/Micro	Hybrid	●○○
Conforto and Amaral (2016)	Macro/Meso/Micro	Hybrid	●○○
Klein (2015)	Macro/Meso/Micro	Agile	●●○
Albers et al. (2019b)	Macro/Meso/Micro	Agile / Hybrid	●●○
Kantelberg (2018)	Macro/Meso/Micro	Agile	●●●

mation), which is implemented using a specific technique, such as a task board. As Klein focuses on providing agile practices for the agility classes, he has captured values as a means to transfer them into a large set of agile practices. As the methodology aims to provide guidelines on how to implement agile practices in machine and plant engineering, the values are a means to arrange these practices, but they are not a vital aspect of his work.

Albers et al. (2019b)

An approach that focuses predominantly on a macro-logical description of development processes is the **Agile Systems Design** (ASD) approach. It aims at supporting *"flexible development in a project-specific, situation- and demand-oriented manner [which] should include mechanisms that, on the one hand, enable the classification of the respective development situation with regard to its planning stability and, on the other hand, provide practices that meet the requirements and demands of mechatronic system development"* (Albers et al., 2019b, p. 1018). The result of this work is twofold: first, it provides a model that allows a team to assess their individual problem situation in terms of entropy and thus planning stability by the use of an "entropy compass" (Breitschuh et al., 2018). Secondly, nine principles are derived from literature and empirical investigations that set out a system of objectives, and these nine basic principles *"serve as guidelines to align activities with them to identify, develop and adapt practices that support developers in the product development process"* (Albers et al., 2019b, p. 1018). The principles are of normative nature, yet they are very generic and therefore not as concrete and precise as values (cf. ASD principle 1: the developer is the center of product development, or ASD principle 2: Each product development process is unique and individual).

Kantelberg (2018)

The **methodology for designing agile product development processes** is a holistic approach that aims at aligning the benefits of agile software development to technical product development. The methodology builds on the ZHO model[12] and consists of four partial models. In the initial step, a shared understanding of product development processes is created. Next, the working mechanisms of ag-

[12] Goal system, action system, object system. In German: ZHO = **Z**ielsystem, **H**andlungssystem, **O**bjektsystem.

ile software development are decomposed into modes of operation. These modes of operation are then aligned to the context of technical product development, which is the basis for the actual modeling of processes for agile development of technical products in the last step. The methodology is modeled thoroughly, as it examines a vertical integration on the macro-, meso-, and micro-level as well as a horizontal integration regarding people, environment, and process. However, two aspects differentiate Kantelberg's take from the approach chosen in this work: As Kantelberg has used the ZHO model, the overarching focus in his "goal system" is on maximizing productivity, which he defines as the sum of increasing both effectiveness and efficiency in the development of technical products. In this line, in the second step, Kantelberg has chosen only three agile process models (from the agile software period) and compared their aims to these of the Manifesto. Due to these boundary conditions, this methodology incorporates agile elements and procedures; a comprehensive transfer of elements on the normative layer, however, is still lacking.

Summary of 2.3

It can be summarized that all three approaches and methodologies address normative elements, but none aim at specifically aligning plan-driven and agile values. This context-specific alignment is carried out in the work at hand; additionally, three perspectives (or spheres) are taken: (1) agility in software, (2) plan-driven in mechatronics, and (3) agility in mechatronics. In chapter 5, these are considered separately, and the respective values for these perspectives are extracted and aligned into an overall model.

2.4 Sense-making and prospects for agile development

Need for agility

Agility as the capability to respond to changes is driven by the existence of change. The changes are multi-faceted and can either be exogenous or endogenous to a development project (Stelzmann, 2011). When taking the five layers of Hales and Gooch as a reference, exogenous changes refer to the layers of *environment* and *market*, such as changing competitors or even a pandemic. Exogenous changes refer to aspects within the company, which are represented by the layers of *management*, *project*, and *design* in the Hales and Gooch model, such as a lack of knowledge, changing directives, or discarding a concept. The latest version of VDI 2206 states that technological changes due to intense connectivity between the systems, the digitalization of products and services, and the linking to the IoT (Internet of Things) are among the most recent drivers for change. Decreasing product lifecycles and earlier value delivery are in line with constantly evolving technologies that complement the drivers for agility (Tseng and Lin, 2011).

VUCA

In line with the characterizations of change, the **VUCA** acronym is used to describe the circumstances under which agility is most meaningful to apply, which stands for volatility, uncertainty, complexity, and ambiguity. *Volatility* relates to the rate of change and unstable conditions, whereas *uncertainty* refers to a lack of information about whether a particular action will initiate change or not (Bennett and Lemoine, 2014). *Complexity* implicates the vast number of elements and their interconnection in both the product and organizational matters, which might trigger change, whereas *ambiguity* implies that it is unclear what might happen

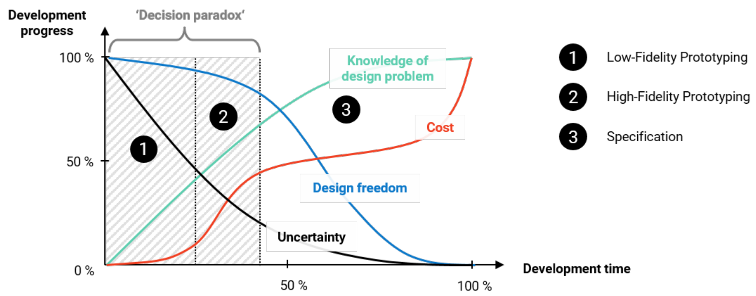


Figure 2.15
Design process paradox according to Ullman (2010).

after a change due to missing cause-and-effect chains (Bennett and Lemoine, 2014). If a development project is exhibited to these circumstances, such a context is referred to as VUCA environment (Bennett and Lemoine, 2014). However, this term was initially coined in the military, which was later adopted by strategic management to relate to fast-paced, ever-changing industries (Bennett and Lemoine, 2014; Schmidt, 2019).

Bennett and Lemoine have investigated such environments in detail and suggest the following strategies to deal with VUCA dimensions. As in the definition by Böhmer et al., flexibility shall be applied to counteract volatility, and a constant gathering of information shall approach uncertainty. An effective means to cope with complexity is to restructure the product and organizational system, which is known as "Refactoring" from agile software development and has already proven to be beneficial. Moreover, experimentation and exploration is a promising approach in addressing and eradicating ambiguity. Agility as a concept, as it is described here, consists of all four strategies and is in line with the understanding of Schmidt (2019).

Design process paradox

In agile product development, the development team is supposed to be cross-functional, meaning that it is staffed with people of different backgrounds and skills. Moreover, agility favors intense communication and a decentralized type of decision-making. However, this is challenging due to diverse development roles within the team and the abstract and uncertain nature of product development in engineering design (Böhmer, 2018). Ullman refers to this as "design process paradox", as displayed in figure 2.15, which shows the insufficient knowledge and lack of information at the beginning of an engineering design project that is needed to make decisions. As the knowledge about the design problem increases over time, the design freedom and potential to influence the design decreases significantly (Freisleben and Schabacker, 2002). One approach to deal with this situation is creating early prototypes to gain insights and learn about the design problem (Ullman, 2010). Early prototypes are a cost-effective and implicit risk insurance that increases the quality of the product by a steep generation of knowledge due to validated learning (Douglass, 2016). By increasing prototyping up-front, expensive and time-consuming errors can be avoided at the end of a project (Punkka, 2012). One fundamental challenge of product development is the continuous balance between the need for early decisions (due to cost pressure) and their delay for the sake of gathering additional information to choose the most effective solution (Schmidt, 2019).

Contextual prerequisites

Regarding the applicability of agile development, certain boundaries need to be established. Böhmer has summarized the requirements for agile product development: Besides using agile process models from a technical-management view, the ability and resources for creating prototypes in a creative atmosphere are of utmost importance. The limitation in terms of time calls for an adequate skillset of the interdisciplinary staffed development team. Moreover, a project setting must be defined, which favors the use of agile process models and enables resource-efficient prototyping. Flexibility regarding the project outline to foster creativity and freedom is necessary to generate feasible or marketable products. Specific roles are necessary, such as a Systems Architect to guide the team towards meaningful and value-adding feature creation. Depending on the complexity of the product to be developed, a suitable approach needs to be chosen which balances the use of agile and systematic (plan-driven) methods (Böhmer, 2018).

Sweet spot

A context description under which circumstances the application of agile *software* development is feasible is given by Kruchten. He refers to such a context as sweet spot, meaning *"the conditions under which most 'labeled' agile software development [principles and practices] have been developed, and for which success is pretty much assured"* (Kruchten, 2011, p. 355). Kruchten summarizes the sweet spot as follows: A small group of ten to fifteen people, co-located, customer availability, interactive business applications, new developments (greenfield projects), RAD programming environment, short development cycles of only a few months, and a common culture (vocabulary and implicit process understanding).

On the contrary, the contexts that cast agile projects into jeopardy are labeled *bitter spot* and have the following characteristics according to Kruchten: Large development teams (groups), distributed development teams, no empowered customer representative to give feedback and resolve issues by negotiation, long cycle times, inefficient development environment, and different development cultures (Kruchten, 2011). A thorough list of the sweet spot conditions for agile development can be found in Schmidt (2019). However, Kruchten mentions that the current methods were designed for sweet spot contexts, yet it might be possible to apply the concept of agile development in other contexts as well.

Advantages and disadvantages

By concluding the chapter on agile development, a brief summary on agile development is presented in table 2.4. Reports from practice present manifold benefits, hence this summary shall contrast the advantages of developing in an agile manner compared to its reported disadvantages (Lévárdy and Browning, 2009; Tseng and Lin, 2011; Kruchten, 2011; Ovesen, 2012; Link, 2014; Schmidt et al., 2018b, 2019).

Studies show that agile development shows significant benefits regarding soft factors, such as *communication* among team members, shared *commitment* towards the agreed goal, or increased *transparency* (Atzberger et al., 2020a). On the contrary, hardly any benefits are seen regarding the so-called hard controlling key performance indicators (KPIs) *quality*, *cost*, and *time*. Regarding quality and time, scarce benefits have been reported over the past three years (Schmidt et al., 2018b, 2019; Atzberger et al., 2020a). Few sources report a significant reduction

Table 2.4 Advantages and disadvantages of agile development based on Schmidt (2019).

Advantages	Disadvantages
✔ Fosters quick adaptation to evolving changes by, e.g., self-organization, team ownership, and decentralized decision-making.	✖ Even though it is well-established in software industry, the application of agile development outside the realm of software is still quite difficult.
✔ Uncertainty is not perceived as a risk to be managed, but as promising opportunity to be exploited.	✖ Taking inherent risks requires courage and self-confidence.
✔ By iterating in a steady rhythm, agile development has short learning cycles and takes advantages of it in favor of the customer.	✖ Team ownership and corresponding empowerment can cause a loss of influence and power of managers.
✔ Frequent generation of prototypes ensures that the project stakeholders have the same understanding of the product. Due to frequent stakeholder involvement in the process, they can actively shape the product. This avoids the big bang situation at the end of the project that often results in comprehensive rework.	✖ Agile development might risk to explore more than needed which results in unnecessary costs. Balancing exploration and exploitation in advance is challenging and can usually be assessed in retrospect only.
✔ Delivering customer value is one of the highest priorities of agile development, as it correlates with customer loyalty.	✖ Multi-skilled employees are needed for agile development.
✔ By striving for potentially shippable prototypes, agile development induces customer value early because the user could use the prototype for productive purposes even when lacking certain features.	✖ Agile development cannot be applied in a checklist manner as it heavily depends on an adequate mindset (cf. the way of thinking described in the Manifesto).
✔ Agile development avoids the throw-it-over-the-wall attitude by calling for interdisciplinary teams with members from different domains for the whole project duration. This increases the commitment, motivation, and identification with the product of the team members, and underpins the important role of tacit knowledge in innovation.	
✔ Agile development aims at keeping overhead workload and bureaucracy at a minimum in favor of creativity and innovativeness.	

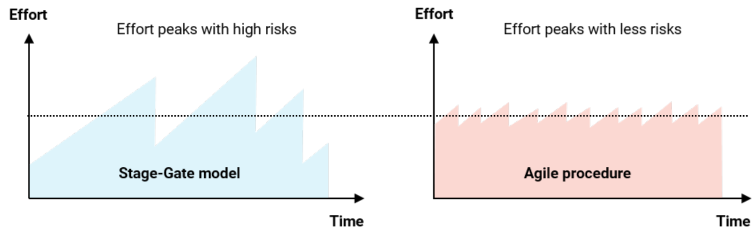


Figure 2.16
Effort peaks between classical and agile ways of working according to Böhmer (2018).

in development time (Klein, 2015; Kantelberg, 2018), as they argue that developing in an iterative manner generates faster insights and thus shortens product development time visualized in figure 4.6 in chapter 4. In this manner, an expert noted that *"even though this is the case, one might not underestimate the time for communication and coordination, which counteracts this effect"* (Interviewee G). The fact that agile development is hyped combined with the guru problem might lead to such valuations (Atzberger et al., 2019b). Regarding development costs, no benefits are reported; the opposite is rather the case due to allegations of frequent prototyping (Atzberger et al., 2020a). The short-cyclic (iterative) and incremental procedure in agile development show its benefits in decreasing effort peaks and a more continuous workload, as visualized in figure 2.16. Such a decrease in workload has also been reported in Atzberger et al. (2020a).
Böhmer reports that *"iterative development and continuous reviews on the project progress allow smoother effort with fewer risks over time. Early prototypes ensure feasibility and facilitate user-tests, to reflect on added user-value"* (Böhmer, 2018, p. 137). Further, she claims that their focus on "delighting the customer" instead of maximizing shareholder value is what makes organizations like Apple, Google, and Tesla so successful, as making money is not their primary concern but the result of their customer-centric actions. This customer-centric approach is one of the key concepts of agile development. An agile team aims to understand the user context by frequently questioning why to explore the actual underlying user need that needs to be fulfilled and find a technical solution accordingly (Schmidt, 2019). As Ries refers to, this constant validated learning is what distinguishes agile from classical development. The systematic study and scrutinizing of unknowns by constantly reframing the problem and its solution characterizes agile development (Ries, 2011). In contrast, classical development builds on the knowledge present at the beginning of the project and intends to reach the projected target in a single well-thought-out approach (Nerur and Balijepally, 2007; Schmidt, 2019). Thus, agile development incorporates frequent feedback loops, leading to insights and opening up possible (unplanned) approaches towards the intended goal (Schmidt, 2019).
Overall, agile development shows a broad range of advantages when applied correctly and in the appropriate context. Important in this regard is that the Manifesto for Agile Software Development and many of the agile process models have been come up with by practitioners from software development. The extent to which this context alignment shows an impact in applying and adapting to other fields is described in chapter 4.1 in more detail.

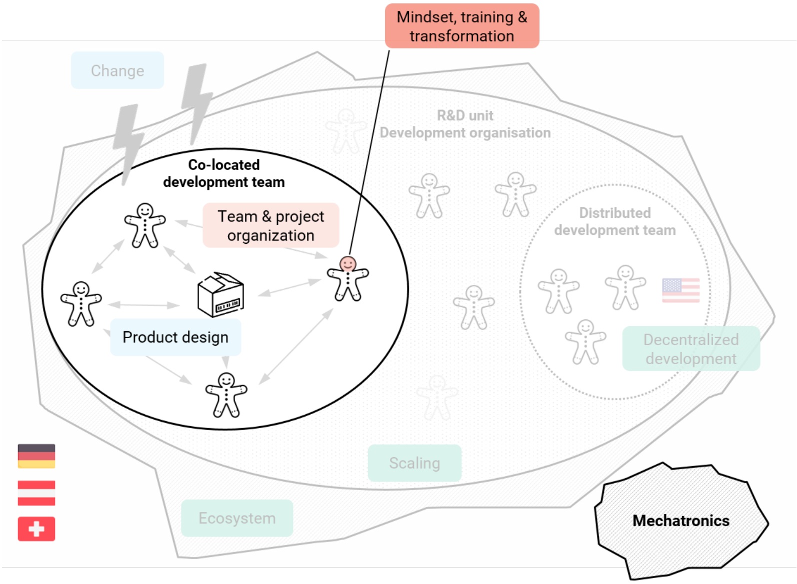


Figure 2.17
Detailed description of the area of investigation.

Summary of 2.4

The state of the art is concluded by a short summary of the scope of this dissertation. The work focuses on the development process, which is located in the product lifecycle. As classical product development processes struggle under VUCA conditions, this work aims at investigating **agile product development processes**, which are flexible and adaptive. These are investigated in the context of **mechatronic products**. Since agile working methods are gradually finding their way from the software domain into mechatronic development, the aim is to align the modes of action at the normative level. In order to achieve this, the **values** and value concepts of agile development in mechatronics are elicited. Since values differ locally, the geographical boundary is limited to the **DACH region**. The focus of this thesis is highlighted in figure 2.17.

3

Research approach

In this section, the research approach is explained in more detail. First, the research methodology used in this dissertation, the Design Science Research (DSR) approach, is elucidated in *chapter 3.1*. A generic procedure of how research is carried out with this approach is shown to display how the DSR is applied in practice. Following that, the Design for Values (DfV) approach is outlined in *chapter 3.2* since it is the main approach upon which the progress in this dissertation is built. Besides elucidating the approach, its application is exemplified by a use case from literature, highlighting the strengths and relevance of this approach.
Chapter 3.3 gives an overview on how DSR is applied in this dissertation. Therefore, the different steps and artifacts which have been generated here are aligned to the generic sequence of the DSR. This enables the reader to comprehend how the research was carried out and how the different artifacts are inter-connected. In line with this, the second part of the chapter displays the methods used and the data generated by the application of these methods. As a mixed-methods approach was applied in this research, all relevant information regarding the method application of both qualitative and quantitative methods is given. Since the research draws heavily on empirical data from experts in the field of agile development in mechatronics, further information on the contextual background of these experts is provided.
Lastly, *chapter 3.4* summarizes the results and gives an outlook on how the following chapters fit into the overall structure.

3.1 Design Science Research methodology

Design Science Research fundamentals

The term "Design Science Research" (DSR) was first mentioned by Hevner et al. in 2004. In a revised version, they summarized what characterizes DSR by describing what is meant by *design* and what distinguishes *research*. On this basis,

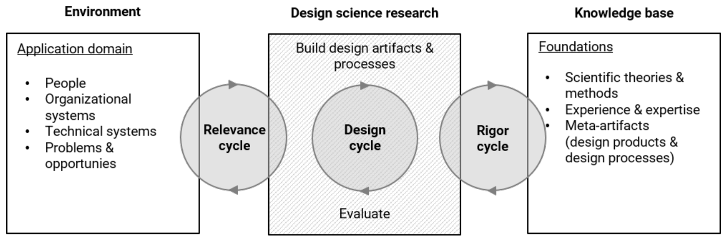


Figure 3.1
Design science and its corresponding components according to Hevner (2007).

DSR is defined as follows: *"Design science research is a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing new knowledge to the body of scientific evidence. The designed artifacts are both useful and fundamental in understanding that problem"* (Hevner and Chatterjee, 2010, p. 5). They substantiate this definition by explaining why it is important to generate fundamental design knowledge and why this science of design is crucial: *"There are theories and design principles in individual design domains such as architecture, engineering design, and software engineering. But a science of design will not emerge from core domains. It has to come from an overarching disciplinary scientific field. The science of design and its theories should be generalizable and applicable across a wide variety of domains and specialties"* (Hevner and Chatterjee, 2010, p. 4).

At this point, an exhaustive discussion on positivism vs. constructivism could follow. However, these two views have already been contrasted vividly by Ovesen in the context of agile hardware development (Ovesen, 2012, p. 53ff.). He concludes this discussion by pointing out that, from an engineering perspective, the investigations usually come from the natural sciences and are thus often rather positivistic. However, since it is a cross field, design engineering utilizes both constructivist and positivistic approaches in scientific practice (Ovesen, 2012). Although one of the two approaches usually has the upper hand in academic work, the DSR allows both sides to be considered equally.

DSR in a nutshell

Design science research suggests that reference should be made to the application environment, the knowledge base, and design science research. These areas are linked by three inherent research cycles: the relevance cycle, rigor cycle, and design cycle, as shown in figure 3.1. They are described as follows in Hevner and Chatterjee (2010): As good design science research often starts by identifying opportunities in an application environment, which it aims to improve by introducing new artifacts, the relevance cycle initiates design science research by providing requirements for the artifact to be developed as well as acceptance criteria for the evaluation of field testing. As DSR draws from a vast knowledge base of scientific theories and methods, the rigor cycle provides a thorough grounding of the research in existing knowledge and therefore ensures that the designs produced are research contributions. Lastly, the design cycle is the connecting link as it iterates between the construction of a design artifact, its evaluation, and subsequent feedback to refine the design. The interaction of these three cycles is the basic idea of the DSR. The effort between creating and evaluating the design artifact must be balanced so that both the academic validity from the rigor cycle and the practical relevance and acceptance of the relevance cycle are ensured.

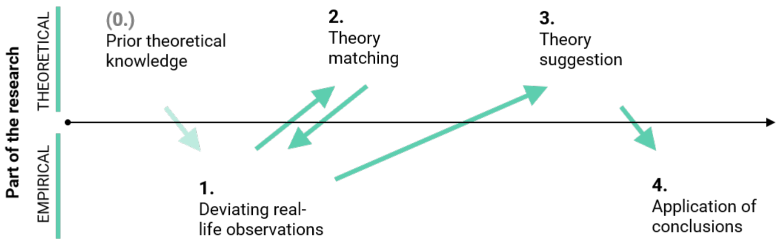


Figure 3.2
Abductive research process according to Kovács and Spens (2005).

The procedure in the design science research procedure of the approach used in this dissertation is presented in figure 3.6 accordingly. A detailed description of the cycles and their interplay can be found in Hevner and Chatterjee (2010).

DRM ↔ DSR

In contrast to the design research methodology by Blessing and Chakrabarti which provides a guideline on how to proceed based on its four stages, the DSR does not initially specify which stages are to be followed. Due to a high degree of uncertainty throughout the initial investigations in this thesis, the DSR turned out to be very practical as it is a flexible approach, thus the research advances and matures with increasing progress in knowledge. Although other "design science / research" approaches also permit open-ended research, the DSR has already been used to investigate a topic on agile development by Schmidt and proven to be a decent fit.

Remark

It should be noted that, in retrospect, the procedure in this work does not differ vastly from that in the DRM. Nevertheless, from the author's point of view, a less prescriptive research procedure such as the DSR is very similar to agile ways of working and, therefore, suitable for such an ambiguous topic. Since the problem definition in this work was a particular challenge, the approach in the DSR proved to be very practical, as it would not have been expedient to define a specific area of investigation at the beginning.

Contribution type

The research strategy chosen in this thesis is *abductive*. In comparison, an inductive research process aims to infer the generality from empirical observations of individual cases, whereas a deductive approach attempts to derive the individual case from precise theoretical assumptions (Kovács and Spens, 2005). An abductive approach builds on existing theory and adds new empirical data to generate new theories. This is not a linear process but an iterative interplay and often seen as a combination of a deductive and an inductive approach (Kovács and Spens, 2005; Ovesen, 2012). Therefore, the research method may vary between iterations, which fits the pragmatic approach of design science (Schmidt, 2019). The abductive research procedure is illustrated in figure 3.2 by the alternation between empirical and theoretical research components.

Summary of 3.1

The work at hand uses the DSR approach and is an abductive research process. The alignment of the individual research steps in this work is displayed in chapter 3.3.

3.2 Design for Values

The Design for Values approach is a subset of Value Sensitive Design, which will be described briefly at the beginning prior to an analysis of the Design for Values approach and its methods.

Value Sensitive Design

Aligning societal and moral values into the development of new technology has its origins in the 1970s at Stanford (van den Hoven et al., 2015). Since then, it has been adopted by several research groups and is most commonly referred to as "Value Sensitive Design", a term that Batya Friedman has coined (Friedman, 1997; van den Hoven et al., 2015). *"Value Sensitive Design (VSD) represents a pioneering endeavor to proactively consider human values throughout the process of technology design"* (van den Hoven et al., 2015, p. 11). Thence, *"key to value sensitive design is its basis in an interactional understanding of technological appropriation. [...] Values are not embedded within a technology; rather, they are implicated through engagement"* (van den Hoven et al., 2015, p. 15). The methodology behind VSD is a tripartite one, consisting of conceptual, empirical, and technical investigations (Friedman and Khan, 2003).

In the *conceptual* investigations, the relevant direct and indirect stakeholders of the technology under study are identified. Additionally, the second step of conceptual investigations includes the identification and definition of values, which are implicated by the use of a particular technology. Here, an example from van den Hoven et al. is taken, in which a company that is building security equipment for houses in Vancouver, Canada, has to define what the value of privacy means in the respective context due to the fact that privacy is a legislated right in Canada. In this respect, value conflicts come into play since it is necessary for the company to balance the residents' conceptualization of security compared to the privacy of other individuals entering the building (van den Hoven et al., 2015).
Following that, *empirical* investigations examining stakeholders' "understandings, contexts, and experiences" in relation to the technology and the implicated values shall be conducted (Friedman and Khan, 2003; van den Hoven et al., 2015).
Concluding the methodology are the *technical* investigations, which are primarily concerned with specific features of the technologies, and which *"include [the act of] designing a new technology in order to support particular values or analyzing how particular features of existing technologies implicate certain values in a context of use"* (van den Hoven et al., 2015, p. 16). This last part can be summarized as to what engineers refer to as the engineering design process, with requirements that incorporate and explicate specific values. For more theoretical information, see Friedman and Khan (2003), and its practical applications of the tripartite methodology are elucidated in Friedman et al. (2008).

Critiques on VSD

As with every approach, the VSD is facing certain critiques as well. These can be summarized as universal values (values being too generic for practical application), ethical commitment (towards these values), stakeholder participation, and the emergence of new or additional values, among others. These are described and contrasted in van den Hoven et al. (2015) more precisely.

In line with the VSD, several other research groups have picked up on it and developed similar approaches, such as *values in design* at the University of California (Bowker and Star, 2000), *Values and Design* at New York University (Nissenbaum,

2001) or at the Delft University of Technology, where they refer to it as *Design for Values* (van den Hoven, 2007). At its core, these value-oriented approaches go beyond the traditional perspective of the engineering design process and aim to *"help researchers and designers explicitly incorporate the consideration of human values into their work"* (van den Hoven et al., 2015, p. 12).

Jeroen van den Hoven, Head of the Delft Design for Values Institute, has given a vivid example to demonstrate how incorporating values into the engineering design process (or the lack of doing so) affects the use of a product.

Example of value consideration

In 2008, the Netherlands have decided to introduce smart electricity meters in the households of its inhabitants nationwide. A smart electricity meter is a small device installed inside a house that monitors the house's electricity consumption. The smart meters should make the electricity grid more efficient and sustainable by balancing the load on the grid due to the real-time data gathered. By implementing smart meters nationwide, the goal was to take a proactive step towards meeting the EU CO_2 emission reduction targets by 2020. After several years of research and development, privacy concerns arose as the device was seen as a 'spying device' and, therefore, a threat to the personal sphere of life. By the time the proposal was presented to the Dutch parliament for approval, public concerns about the privacy of the device had risen significantly that the whole project has been rejected due to privacy data projection reasons (van den Hoven, 2021).

This example illustrates the importance of incorporating values into the design process and the continuous monitoring of these values, especially since data privacy and privacy concerns by sharing personal data were still relatively new to the public ten years ago.

Design for Values

In the following, the Design for Values (DfV) approach is explained in more detail. The *traditional view* of design is that it *"is a technical and value-neutral task of developing artifacts that meet functional requirements formulated by clients and users. These clients and users may have their own moral and societal agendas, yet for engineers, these are just externalities to the design process"* (van den Hoven et al., 2015, p. 1). In contrast, the DfV approach, which has been developed in Delft, focuses on incorporating values into the discipline of engineering and, thus, the engineering design process. Ibo van de Poel, one of the leading experts of the DfV approach, defines *engineering* as an *"activity that is aimed at understanding, creating, improving, and maintaining certain technologies"* (van de Poel et al., 2010). In this context, three value-related perspectives on engineering will be considered in more detail, which the DfV scholars have identified.

Value-related perspectives

For one, *engineering as a profession* explains how various engineering professions also have different value sets, which they respect. Given the area of chemical engineering, the central values in that discipline are, e.g., *physical safety* and *health*, whereas in software engineering, such values only play a minor role. There, the value of *privacy* plays an important role, which is rather irrelevant in chemical engineering (van den Hoven et al., 2015). Even though some of these values are rather specific to a particular domain, some values are shared among most engineering professions (van den Hoven et al., 2015).

Second, *professional codes in engineering* are aspirational and describe the core values of the engineering profession. ASME once formulated professional codes which state that *"engineers should hold paramount to the safety, health, and welfare of the public"* (van den Hoven et al., 2015). In this light, the most modern professional codes relate to conducting a profession with honesty and competency, obligations towards employers and clients alike, and responsibility towards the public and society (van den Hoven et al., 2015). These codes shall therefore be considered as honorable or ethical codes of a profession.

Third, as previously mentioned, the aim of both Value Sensitive Design and Design for Values is to incorporate moral values in the engineering design process. This implies that there are different types of values and also non-moral values. Since *"the differentiation of what is referred to as a moral or non-moral value is a philosophical one"* (van den Hoven, 2007) and not of importance for this investigation, the differentiation is made between *instrumental* and *intrinsic (final) values*, which are of importance for the work at hand. The latter are values that strive for their own sake, whereas instrumental values strive for the sake of other values (van den Hoven et al., 2015). However, there is disagreement between philosophers about which value is considered instrumental or intrinsic.

⚠ Definition

To settle this issue, the thesis at hand follows the classification of the DfV scholars. According to them, *"values like human well-being, justice, safety, health, and sustainability that play a major role in engineering are **final values**, while values like economic profit, efficiency, reliability, and maintainability, which are obviously also important engineering, are **instrumental values**"* (van den Hoven et al., 2015, p. 669). This distinction can be illustrated with the example of the smart electricity meters mentioned earlier; here, an instrumental value is the efficient use of electricity by installing it in households nationwide. The final value in this example is sustainability, i.e., the ecological use of limited resources.

It is important to mention that no list exists that exhibits all values that are relevant for the discipline of engineering. However, it is possible to identify specific sets of values for different branches such as information technology, architecture, or vice versa, to explicate a particular value based on the Design for X approach, meaning that when, e.g., designing for sustainability, all relevant aspect in one domain or across multiple domains could be gathered. These approaches are further elaborated in several chapters in van den Hoven et al. (2015). In the next paragraph, the translation of abstract values into concrete design requirements is illustrated.

Values hierarchy method

One method that is of great importance for the DfV approach in general and the thesis at hand in particular is the values hierarchy method, which is displayed in the form of the values hierarchy pyramid in figure 3.3. It is divided into three levels, the top level containing both intrinsic and final values, the middle level containing norms/objectives, and the bottom level containing design specifications. When applying this method, all values that are important to an engineering design project are located at the highest level.

It is mandatory for a company to determine an initial set of values representing the company's values and beliefs first. Next, management has to determine which final and instrumental values are relevant for a given design project and therefore need to be taken into consideration. Based on that input, suitable values can be chosen, and additional ones might even be added, if necessary or

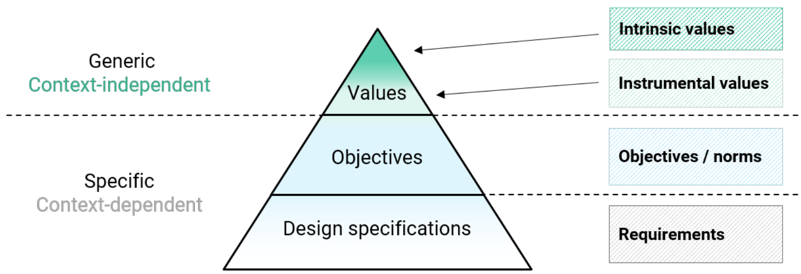


Figure 3.3
Values hierarchy method according to Delft Design for Values Institute (2019).

meaningful (van den Hoven et al., 2015). In the next step, a generic value needs to be translated into one or more general objectives represented by the intermediate level of the values hierarchy.

Van de Poel highlights that **norms** stand for all kinds of prescriptions and restrictions that guide our actions (van de Poel, 2013). One kind of norms are so-called "end-norms" which refer to an end to be achieved or strived for (van de Poel, 2013). These end-norms may refer to properties, attributes, or capabilities of the artifact to be designed and may include what is referred to as **objectives** (like "maximize safety" or "minimize costs" without a specific target) or goals (that specify a target such as "this car should have a maximum speed of 120 km/h") (van de Poel, 2013). Hereafter, these general norms have to be translated into more specific **design requirements**, representing the most concrete and specific level of the hierarchy. The process of translation is referred to as specification or operationalization in this context (van de Poel, 2013). Therefore, the values hierarchy method links every value to one or more objectives (norms) and subsequently design specifications and thus, makes its interrelations explicit.

Figure 3.4 shows the functionality of the values hierarchy in a concrete example, the construction of cages for hens (Delft Design for Values Institute, 2019).

Example:
Chickens in the EU

The example deals with the development of cages for chickens in accordance with an EU directive. Important to mention here is that values are, by definition, context-independent due to their generic nature. In this example, animal welfare and human well-being are listed as values, but also other values could be taken into account, including environmental sustainability, safety, or transparency (as an instrumental value; subordinate to sustainability as a final value). When being translated (i.e., specified), these values become more explicit since the context in which they are applied comes into play – general norms are therefore context-dependent and more specific than values. The objective of having enough living space per hen and considering an appropriate design of the perches is an explication and thus operationalization of the value of animal welfare. At the bottom layer, as the result of a further specification of a norm, the design requirement is most concrete and context-dependent as well – this is what engineers commonly refer to as a requirement. A concrete number of 1,100 cm^2 useable space per hen is a typical functional requirement for a design project. As indicated in Figure 3.4, a value can be specified top-down, but it can also be generalized bottom-up based on existing requirements. However, generalization and operationalization do not necessarily lead to the same results (van de Poel, 2013; van den Hoven et al., 2015).

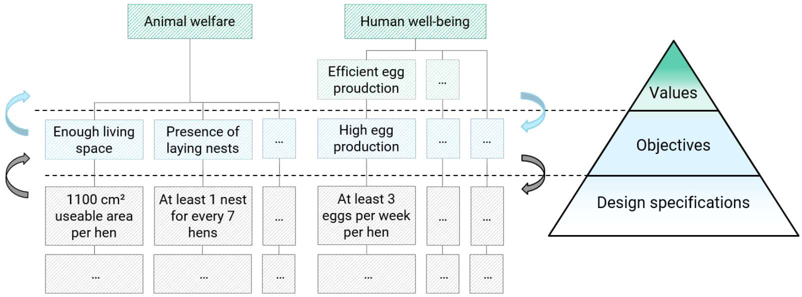


Figure 3.4
Chicken example aligned to values hierarchy method according to Delft Design for Values Institute (2019).

DfV in practice

The key advantage of the values hierarchy is that it is a simple but powerful instrument in the strategic subdivision of a project. The system of values is created, interpreted, and put into action. Even though an integral approach to the DfV approach is still missing, the scholars propose a four-step approach (van den Hoven et al., 2015):

1. Discovery and identification of values
2. Translation of values into engineering characteristics (operationalization)
3. Unravelling conflicting values and trade-offs
4. Verification of these values in the final product

However, the creation is an iterative process that often consists of a combination of specification (top-down) and generalization (bottom-up). The values hierarchy represents a method for clarifying the values, norms, and conflicts present in the design process (van de Poel, 2013). Translations need to be made to operationalize the aforementioned values or objectives in every step. The operationalization of values can never be objective without intermediary second-order value judgments (van den Hoven et al., 2015). The application in practice is discussed in chapter 7.2 in greater detail.

Furthermore, it needs to be mentioned that in addition to the geographical limitation of the validity of values, as already stated in chapter 2.3.2, values can also change over time. Additionally, values reflect needs. Interviewee L mentioned that *"since values are always valid in general, it implies that they are 'accepted' by a large number of people."*

Summary of 3.2

The DfV approach was originally conceived for the development of products. In this paper, the DfV approach is used to study the process of engineering design. The aim is to identify the relevant values that are essential for successful agile development in the context of mechatronics. Interviewee L added that the approach in this work seems coherent as "it is the process that realizes the values."

3.3 Chosen research strategy

While the DSR has been explained in chapter 3.1, the DfV approach has been described in chapter 3.2. In the first part of this chapter, the own research approach is placed in the chosen procedure, whereas in the second part, the methods used and data generated are elaborated.

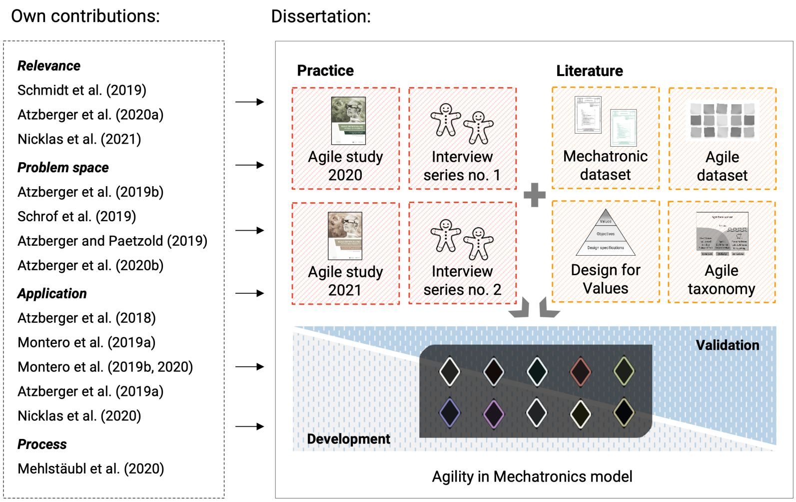


Figure 3.5
Overall course of action and own contributions.

3.3.1 Overall course of action

Preliminary work

The research approach in this work is based on previous findings that have been developed in the context of investigations at the author's institute (ITPE). These findings represent the author's learning curve and show which areas have been investigated and to what extent the findings from these areas have influenced the actual work in this thesis. A total of 14 preliminary papers have been produced as part of the work at ITPE, which can be divided into four areas. The first area, *relevance*, evaluates the current state of agile development in mechatronics by annually recurring studies on its practical application. The second area deals with *problem space* studies that focus on issues and challenges in applying agile development and possible solution approaches and their effects. The third area deals with the *application* of agile development in the different fields and analyzes modes of action based on several case study results. Finally, the *process* area investigates models of information and data flow modeling in product development. A detailed overview and further information regarding intentions, contributions, and methods applied are broken down in detail in appendix A.1, which is why only a brief description of the topic areas is presented here.

Classification in the overall course of action

The own preliminary works and the data generated within the scope of this thesis are displayed in figure 3.5. This representation is intended to serve as an overview and divides the elements used and generated into the practical and theoretical parts. Interviews and studies are part of the practical investigations, whereas multiple datasets of agile and mechatronic process models and scientific approaches (DfV approach and agile taxonomy[1]) stem from literature. All of these elements have been developed or applied within the course of this thesis.

Alignment to DSR approach

In order to comprehend the interaction of the individual elements and the further development in a reasonable way, the same elements have been placed into the DSR approach. The result is a representation of the procedure in figure 3.6,

[1]The term *agile taxonomy* is the abbreviated version of *Taxonomy for agile development* (Atzberger et al., 2020b).

assigned to the respective cycles according to Hevner.
Several works precede this thesis, and various findings have been incorporated into this work. The starting point is referred to as *agile study 2020*[2], which investigated the current state of agile development in mechatronics in German-speaking countries. In the context of this study, the actual area of investigation of this thesis has been validated, namely the problem area regarding the mindset within agile development. Hence, the problem statement represents the first design artifact within the design cycle and is the foundation of the problem description of chapter 4. With the concretization of the problem space, possible solutions have been sought in the literature. The *agile taxonomy* and the *Design for values* approach were identified in the rigor cycle and reconciled with each other. On this basis, the further research procedure has been planned, which consisted of a literature review, from which *datasets on agile and mechatronic process models* have been extracted, as well as an interview study (*interview series №1*). The results from both the relevance and the rigor cycle led to a first version of the Agility in Mechatronics (AiM) model. Based on additional input in a second interview series (*interview series №2*), a revised version of the AiM model has been created. In the last step, the elements of the AiM model have been validated in a broad quantitative study in the course of conducting *agile study 2021*[3], the successor of *agile study 2020*. As a result, the validated AiM model is a design artifact which is now incorporated into the knowledge base as a result of this dissertation.
It can be summarized that the approach in this thesis relies mainly on the experiences of experts from the field of interest. Therefore, a constructivist approach (cf. chapter 3.1) is taken, which generates significant knowledge through insights obtained in empirical research.

3.3.2 Methods applied in embedded design approach

In the following, the actual methods that were used to create the aforementioned data and the design artifacts will be explained. First, the chosen mixed-methods approach is briefly discussed before the methods, and the resulting data are described in more detail in the next step. Here, the methods for model creation are mentioned first, followed by the methods for model validation.

Embedded Design approach In addition to the comprehensive overview stated above, the actual research procedure is outlined in this chapter, as the research design sets out which concrete methods have been used for collecting, analyzing, and interpreting the data. In this thesis, a mixed-methods approach has been identified to fit best the needs of the investigation, i.e., a combination of qualitative and quantitative methods. An investigation regarding the topic of mindset needs to be assessed by a magnitude of experienced participants in the field to ensure its validity and acceptance, which is commonly carried out by quantitative means. The actual investigation of such an opaque topic can only be carried out with highly experienced practitioners through qualitative research. According to Creswell and Plano Clark, there are many types of mixed methods that can

[2] The term *agile study 2020* is used as a synonym here for the publication of Atzberger et al. (2020a).
[3] The term *agile study 2021* is used as a synonym here for the publication of Nicklas et al. (2021).

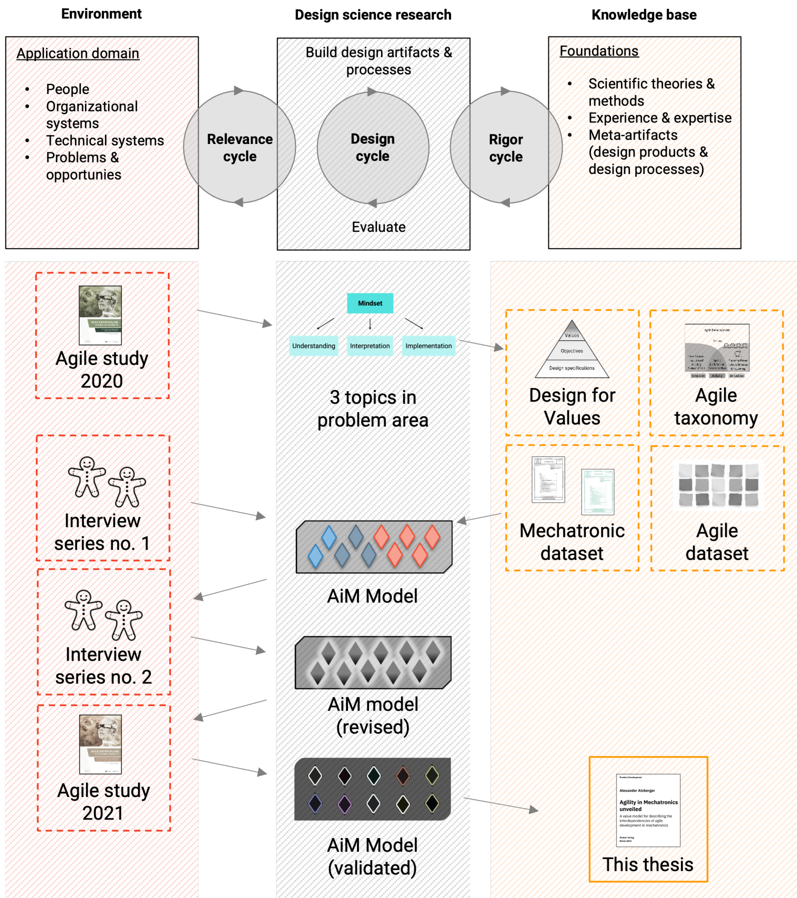


Figure 3.6
Course of action arranged in the Design science research approach.

be divided into six main groups: the convergent parallel design, explanatory sequential design, exploratory sequential design, embedded design, transformative design, and multiphase design. The different design types are compared according to their variants, timing, weighting, and mixing as basic differentiation criteria in (Creswell and Plano Clark, 2007, p. 85).

For this thesis, an embedded design approach was chosen. It can be summarized as a mixed-methods approach where the researcher combines the collection and analysis of one type of data (quantitative or qualitative) within a traditional qualitative or quantitative research design, thus embedding one type into the other (larger) type of research design (Creswell and Plano Clark, 2011). The collection of the second dataset may occur before, during, or after the implementation of the data collection and analysis of the larger design type. It can either be explorative when being collected before the larger design or supportive when being collected during or after the larger design type (Creswell and Plano Clark, 2011). A summary of the different design type variants regarding possible application purpose and mixing strategies can be found in (Creswell and Plano Clark, 2011,

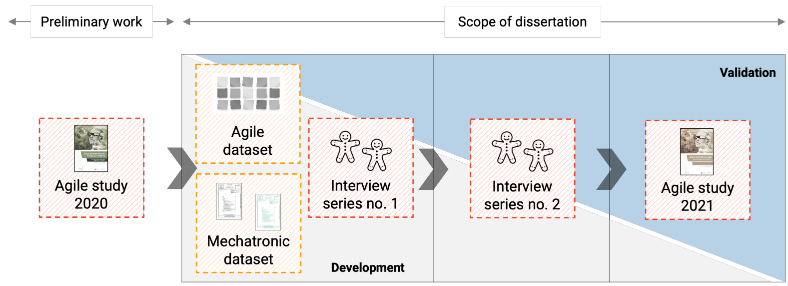


Figure 3.7
Embedded design approach in this dissertation based on Creswell and Plano Clark (2011).

p. 73ff.). In this thesis, the primary dataset comprises qualitative data, which is supported by a quantitative strand, which has been collected at the end of the investigation, thus being supportive to ensure the validity of the data. The embedded design approach with the corresponding methods is displayed in figure 3.7.

Preliminary investigation

The starting point for examining the topic is based on findings obtained in *agile study 2020*, as indicated in figures 3.6 and 3.7. From a methodological point of view, the findings are obtained using an online questionnaire and presented in the format of a scientific study. As these studies are part of an annual study series, they aim to record the status quo of agile development of physical products in the German-speaking region[4]. The survey covers several main topics, and some of the questions are confirmatory to test hypotheses and thus present the current state of affairs. In contrast, others are exploratory to capture inconsistencies or ambiguities (between theory and practice) and identify additional potential topic areas. Since part of these results form the foundation of the problem definition, a detailed description of the results is presented in chapter 4.

Methods used for model creation The creation of the model, which is described in more detail in chapter 5.1, was based on a *complex literature review* and *expert interviews*.

Complex literature review

The *complex literature review* follows the six-step procedure according to Machi and McEvoy (2016)[5], which has been adapted to the research procedure. The literature review aimed to present an overview of the current state of process models for both agile development and mechatronics development (using the categories as already specified in chapter 2.3.3). Based on the selection of suitable process models for both data sets, the value extraction procedure (for each process model) was carried out in the next step. The selected process models for agile development and mechatronics development are listed and discussed in chapter 5.1. The actual procedure for the value extraction is described in chapter 5.2.

[4] Given these constraints, the number of potential participants is already limited. Nevertheless, every survey participant is categorized and thus checked in the demographics section with regard to their experience to ensure that the sample only consist of eligible participants. The exact methodological procedure can be found in Atzberger et al. (2020a).

[5] As Machi and McEvoy point out, a complex literature review is used in every dissertation to outline the research interest (chapter 1) and the subsequent review of the literature (chapter 2), as it is done in this dissertation as well.

Name	Position	Domain	Experience with agility
Interviewee A	Agile Coach	Development	8 years
Interviewee B	Project leader	Development	6 years
Interviewee C	Referent development	Advance development	5 years
Interviewee D	Agile Coach	Production	5 years
Interviewee E	Product Owner	Development	2 years
Interviewee F	Product Owner	Development	2 years
Interviewee G	Head of project management	Development	9 years
Interviewee H	Systems Engineer	Development	7 years
Interviewee I	Agile Coach	Development	6 years
Interviewee J	Research assistant	Research	4 years
Interviewee K	Research assistant	Research	2 years
Interviewee L	Managing Director	Consulting (Development)	12 years
		Average:	**5.6 years**

Table 3.1 Overview of the participants from interview series №1.

In addition to the two data sets from the literature research, interviews with experts have been carried out. Two series of *expert interviews* were conducted (interview series №1 and №2), which differ significantly from each other. Interview series №1 served to create another dataset from empirical research and had been conducted in an explorative manner to determine how agile development is carried out in the context of mechatronics. These three datasets form the basis for the development of the first version of the AiM model, as described in chapter 5.1. In comparison, interview series №2 was confirmatory and had thus been conducted with a different group of experts later in time.

Interview series №1

A total of twelve people from eleven different companies were interviewed. The interviews were *problem-centered expert interviews with partially standardized guidelines*. For the sake of clarity, the following termini are briefly explained according to the definitions given by Heistinger (2006) and Flick (2017). The interviews had the following characteristics: They were interviews with experts, i.e., these people were representative of a particular group of experts. Furthermore, they were partially standardized guided interviews, i.e., the procedure was not strictly prescribed. The interviewees were given a rough guideline beforehand which explained the major topics of the interview. The interviews were thus semi-structured. Furthermore, they were problem-centered as they aimed at identifying relevant values for agile development in mechatronics. Flick understands problem-centeredness as the orientation of the researcher to a relevant social question. Subsequently, the interviews were transcribed and processed according to the general flowchart of qualitative content analysis according to Kuckartz (2018). Specifically, the transcribed interviews were coded using MAXQDA in an inductive-deductive interplay. The results were exported and further processed and analyzed in Microsoft Excel. In the course of the document at hand, reference is made to some text passages from these interviews. An overview of the participants in interview series №1 is shown in table 3.1.

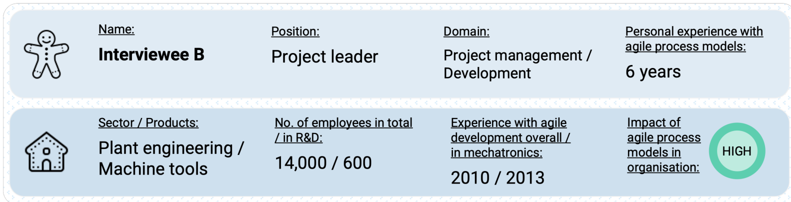


Figure 3.8 Detailed description of the interview participants in appendix A.6 and A.7.

Note: The author is also part of this expert committee.

The participants in this interview series are all members of a VDI[6] expert committee, which was newly founded in summer 2020 and is concerned with the topic of "Agile Development of Mechatronic Systems". Detailed information about the individual participants as shown in figure 3.8 can be found in appendix A.6. The participants of the expert committee are all pioneers in the application of agile working methods in their companies. As the average experience of the participants is 5.6 years, the participants are above-average expertise-wise in the DACH region as of 2020 according to Atzberger et al. (2020a) and therefore considered experts in this field. The data collection took place in september 2020 and was conducted via videoconferences.

The results from the interviews and the literature research form the basis for developing a first version of the AiM model. A detailed description of the data used and its further processing follows in chapter 5. This first version of the model was further refined and thus validated in the following steps.

Methods used for model validation In the course of the validation of the initial model, two methods were found to be meaningful: another *interview series* with experts and a quantitative investigation within the bounds of a *survey*.

Interview series №2

In the second interview series, a total of seven people were interviewed. Since it was already possible to build on an existing model, *focused expert interviews* were conducted according to the definition of Heistinger (2006). Again, the interviewees are experts in this field, and they have even more experience with the topic – 7.4 years on average. Moreover, the structure of the interviews has changed from problem-centric to focused. *"Central to these interviews is the focus on a predetermined conversation topic or 'conversation stimulus' "*, which were the intermediate results shown to and discussed with the experts (Heistinger, 2006, p. 6). Based on their remarks, the results were refined, and the essential normative elements of the model have been confirmed. As this step is neither sheer development nor validation, it is considered as "intermediate step" of both development and validation.

An overview of the participants in interview series №2 is provided in table 3.2. As mentioned above, the participants of this series are highly experienced practitioners. In addition, the selected people are *lead users*[7] in the application of agile working methods in mechatronics according to Schmidt (2019).

[6] Association of German engineers. In German: VDI = **V**erein **d**eutscher **I**ngenieure.

[7] Schmidt used the lead user method in his dissertation and identified the lead users for agile product development in the DACH region.

Table 3.2
Overview of the participants from interview series №2.

Name	**Position**	**Domain**	**Experience with agility**
Interviewee L	Managing Director	Consulting (Development)	12 years
Interviewee M	Agile Coach	Development (Electronics)	8 years
Interviewee N	Head of Sales	Sales & Distribution	5 years
Interviewee O	Agile Coach / Expertise Owner	Advance development	6 years
Interviewee P	Corporate strategist	Organizational development	4 years
Interviewee Q	Head of project management	Development	8 years
Interviewee R	Expert Agile Transition	Development	9 years
		Average:	**7.4 years**

Furthermore, these lead users are part of an expert network called APPe[8], which has been working intensively on this topic for a considerable amount of time since 2016.

Note: The author is also part of this network of agile practitioners.

As these people have the most expertise with the topic, they were chosen as interview candidates to refine and validate the model elements. Due to the fact that all interviewees agreed on the final result of the AiM model, this affirmation represents the first step in the validation process. However, to test the general validity of the model, the individual elements of the model have been confirmed by a large number of people in the field of interest in a second step.

Quantitative investigation

For the second and final step, a quantitative approach was used. For this purpose, five questions were asked in an annual survey on the current state of agile development in mechatronics (referred to as *agile study 2021*, as mentioned above) in order to validate the findings, as displayed in table 3.3. A total of 94-99 people answered the questions (number varies per item, as the option *N/A (no answer)* was not included in the evaluation). Each item was rated on a 5-point Likert scale. In addition to the fundamental elements of the model (M1-M10), three demographic questions (D1-D3) were also recorded. On the one hand, these are important to check the difference between the personal experiences of the survey participant compared to the progress of the respective employer. On the other hand, the question about the perspective is highly relevant since the personal experience with agility can be correlated with one's own assessment regarding the point of view. The detailed results are presented in appendix A.2.

The detailed procedure for data acquisition and further evaluation is explained in more detail in Nicklas et al. (2021). Overall, the individual model components were thus confirmed by more than 100 people involved in the topic. The model as a design artifact will thus enter the knowledge base according to Hevner's DSR approach, where it can be used as a normative aid by further researchers as well as practitioners.

[8] Acronym referring to *agile performance in product development*. In German: APPe = **A**gile **P**erformance in der **P**rodukt**e**ntwicklung.

Table 3.3
Determining questions from the survey of Nicklas et al. (2021) to validate the AiM model, sorted by demographic-related (D) and model-related (M) criteria.

ID	Criteria
	For how many years have you personally been involved with the topic of agility?
D1	not yet, <1, 1-2, 3-5, 6-10, >10 years
	We kindly ask you to define a position (point of view) from which you would like to answer the following questions:
D2	Beginner / learner, contributor, leader, coach, expert
	For how many years has your company been involved with agile working in the following areas? *(not yet, <1, 1-2, 3-5, 6-10, >10 years)*
D3	Software
D4	Embedded software
D5	Physical products
	How important are the following elements for the agile development process in mechatronics? *(Likert scale: 1 = not important; 5 = very important)*
M1	Trust
M2	Motivation
M3	Learning culture
M4	Interdisciplinary collaboration
M5	Continuous improvement
M6	Short-cyclic gain in insight
	With regard to the product to be developed, how important are the following elements for the agile development process in mechatronics? *(Likert scale: 1 = not important; 5 = very important)*
M7	Systems thinking
M8	Connectivity
M9	Safety & Security
M10	Product excellence

Summary of 3.3

A mixed methods approach in the form of an embedded design approach was chosen to answer the research question. The individual research methods have been described in detail to ensure the comprehensibility of the results. Moreover, detailed information regarding the participants of both interview series is display in appendix A.6 and A.7.

3.4 Conclusions on the research approach

Overall, it can be stated that the present work can be attributed to design science. A mixed-methods approach was used, an embedded design approach in specific. At this point, the individual steps of the research approach are summarized and embedded in the overall procedure of the work at hand. Therefore, figure 3.9 has been modified, and the respective chapter notations have been added to give the reader an overview of which chapter deals with the respective datasets.

Chapter 4 begins with a description of the problem, which divides the topic into three subordinate areas. In the next step, chapter 5.1 presents the individual datasets before chapter 5.2 explains the value extraction of the individual datasets. Chapter 5.3 then goes into more detail on the validation steps and describes the

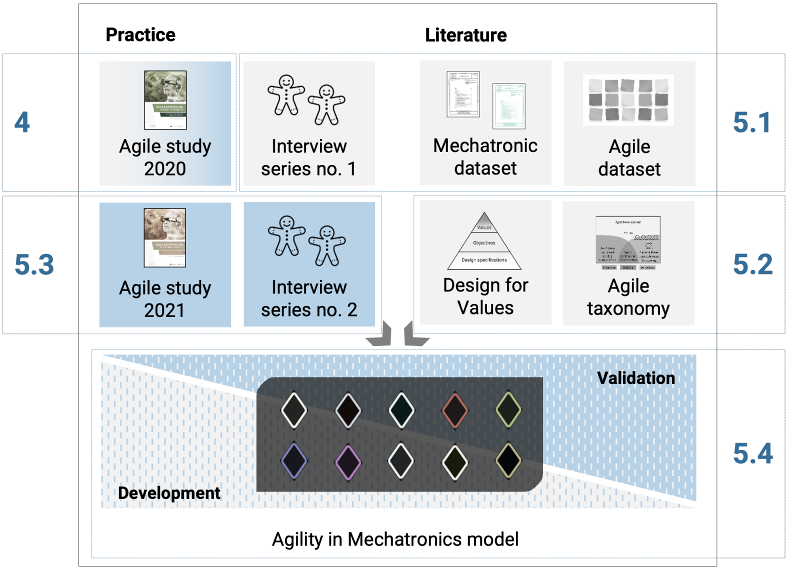


Figure 3.9
Overview of further course of action in the following chapters.

"vehicle" means-end relationships. Finally, chapter 5.4 presents the final AiM model and explains the individual components of the model in detail. In addition, the individual values of the model are compared with the views from Chapter 5.1 and thus contrasted to work out the differences and commonalities.

4

Problem identification & solution approach

Chapter 4 describes the central challenge that is addressed in the course of this dissertation. Therefore, the general issue is explained at the beginning of *chapter 4.1* and specified accordingly in the following sub-chapters.
Chapter 4.2 presents the solution approach chosen and explains why this approach is considered to be appropriate. Moreover, it is elucidated why an appropriate context transfer needs to be carried out by a consistent alignment on the normative layer.
Chapter 4.3 concludes with a consideration of validation criteria which is suitable for the development of models. Thus, the final AiM model is tested against these criteria in chapter 6.

4.1 Problem areas

Research has shown that the application of agile development in software is about ten years ahead of that in mechatronics (Schmidt et al., 2018a). Since agility in mechatronics is thus no longer late-breaking, there are already several experience reports and problem descriptions. Ovesen is considered one of the pioneers who investigated the application of Scrum in seven Danish mechatronics companies and gave a comprehensive description of the associated problems and is thus the reference point (in terms of the region of interest). This investigation

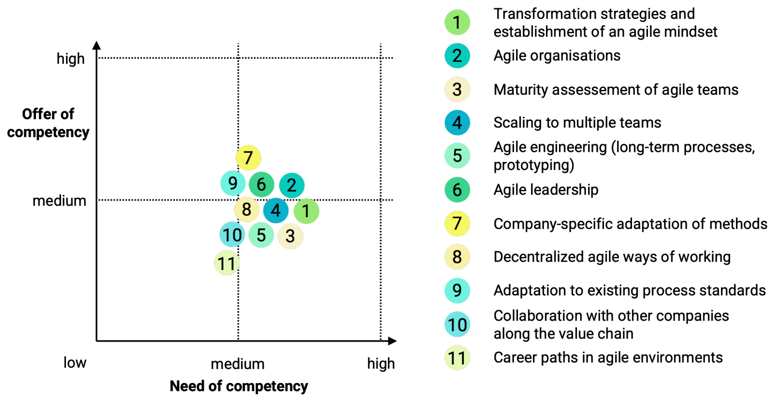


Figure 4.1
Assessment of challenges by experts of the APPe network in spring 2020 (n = 20).

is followed by other studies on specific aspects of that topic, all of which deal with the challenges of applying agile approaches in mechatronics (Berger and Eklund, 2015; Schmidt et al., 2016). Atzberger and Paetzold have updated the original problems as a reference to Ovesen to determine their evolution and advancement.

Although the problems in dealing with the *constraints of physicality (CoP)* were still rated very high, the initial application problems have turned into an issue of comprehension and misunderstanding, which forms the second category and is referred to as *mindset* in this context. These two categories present by far the greatest challenges. The *team distribution* of agile teams (not being co-located) as the third category has gained importance, and the issue of *scaling agile ways of working* has been added as a new field of challenges. Hence, this work examines the issue of mindset in more detail. Within the course of the study series on agile development of physical products, which is conducted annually at the author's institute, particular attention was paid to the topic of mindset in Atzberger et al. (2020a). The topic was specifically addressed in this study to demonstrate its practical relevance and serve as problem validation.

Assessment of APPe experts

In order to further substantiate the practical relevance, an online survey among the members of the APPe network was conducted during a virtual meeting in spring 2020. The aim of the survey, which covered eleven problem aspects, was to identify the needs and the competencies to facilitate the exchange among the network members and to identify relevant topics for the future. Therefore, each aspect had to be assessed regarding the need of the individual member and the person's experience with that aspect. The results of that online survey, in which 20 network members from different companies took part, are shown in figure 4.1. Here, *transformation strategies and establishment of an agile mindset* was rated highest with an average of 7.2 out of 10. This rating underpins the fact that the topic of mindset is still a profound challenge even for the participants of the APPE network (cf. chapter 3.3.2), who are the most sophisticated users of agile working methods in the DACH region. Since the network members have already been dealing with agility in mechatronics for a substantial time (with some of them even being lead users), this result manifests the relevance of this topic for prac-

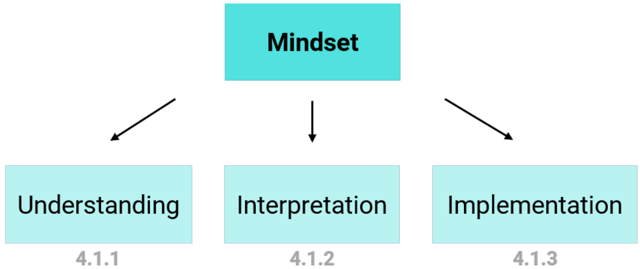


Figure 4.2
Overview of the problem areas regarding the mindset

tice once again.[1]

Mindset as overarching issue

Based on the results of the investigation, three areas regarding the topic of mindset have been identified, as shown in figure 4.2. The areas of *understanding*, *interpretation*, and *implementation* are examined in more detail in the following subchapters. At the beginning of each chapter, the results by Atzberger et al. (2020a) on the corresponding topic area are presented, which are then enriched and supported with additional findings. This detailed presentation is intended to show why establishing an agile mindset is associated with various challenges and why this topic is highly relevant, even for advanced users of agile development.

4.1.1 Difficulties in understanding

The first problem area relates to how agility is understood. As its understanding is manifold, issues arise in its interpretation which inevitably leads to problems in the application of such as a consequence. Therefore, the problem of understanding is first explained in more detail, as well as possible causes for this in the following four aspects.

Variety of attributions

The first dataset from the study on the current state of agile development in mechatronics contains a list of adjectives that the respondents attribute to agility according to their understanding (Atzberger et al., 2020a). The question was asked in an exploratory manner so that everyone was able to state their personal understanding. Each person was asked to name three adjectives, and typos were corrected. The result is shown in figure 4.3 in the form of a word cloud. The size of the terms correlates with the frequency of naming. A total of 95 people answered this question, and a total of 94 different terms were named. At least two people mentioned every term shown here.

However, the three largest terms, *Flexible*, *Transparent*, *Customer-oriented*, were only mentioned by a quarter of all participants. It is also noticeable that a few terms with negative connotations were mentioned. This is due to the fact that agile development is currently hyped (according to the Gartner Hype Cycle) and thus perceived in a negative way by certain people as well (Atzberger et al., 2019b). Nevertheless, it is surprising that among all the people who have been involved with agile development for an average of 3 - 5 years at present (Atzberger et al., 2020a), there is no greater consensus regarding a basic understanding. *"Even*

[1] As lead users are characterized by the fact that they have found a solution to a certain problem before there is an official product launch, the challenges and solutions of this group are highly relevant for the masses, since all subsequent users will encounter the same problems at a later point in time according to the lead user theory (von Hippel, 1986).

Figure 4.3
Most frequent associations with agile development (n = 95) according to Atzberger et al. (2020a).

though this diverse understanding is not a fundamental problem per se, it still bears the risk of misunderstandings" (Atzberger et al., 2020a).

Various definitions

One possible explanation could be the multitude of definitions of agile development in product development. Böhmer examined different definitions of the term *agility* and came up with 16 different variants (Böhmer, 2018, p. 184f.). Since most of these sources come from authors from the domain of software, she thus tried to derive a definition of the term for mechatronics. Albers et al. conducted a systematic literature search in Scopus, particularly for agile mechatronics development. As a result, they selected a total of 14 definitions which, according to the selected keywords, address different perspectives in the period from 2013 to 2018 (Albers et al., 2019a, p. 8f.). This multitude of definitions, some of which address specific sectors or individual aspects, sometimes only represent individual areas and are not holistic, which is why new definitions are often created. Conversely, this does not ensure that readers of different definitions have the same understanding of agility.

Flawsome comparisons

Every comparison needs a reference point against which to compare or measure something. In the literature on agility, comparisons are often made that describe classical, plan-driven approaches as rigid and outdated, as in, e.g., (Nerur et al., 2005, p. 75) or (Schmidt, 2019, p. 18). The jeopardous part of this "agile vs. classic" representation is the categorization in black and white, which is valid from the viewpoint of a model, as it has the reduction feature, but no longer explicitly states this shortening. In their presentation of the Agile Stage-Gate, Cooper and Sommer have softened this extremum by using the term *homeground*, which is a subtle yet crucial difference (Cooper, 2016, p. 13). In conversations with mostly inexperienced practitioners from the agile community, this black-and-white representation leads to a plan-driven approach being characterized as a *waterfall approach* as once formulated by Royce, which is no longer the case nor suitable. However, such abbreviated representations can be found in the literature (Nerur et al., 2005; Schmidt, 2019).

Re-branding of established instruments

Another example that serves as an inadequate representation of the working mechanism of agility is the so-called "Stacey matrix". It is based on the work of organizational theorist Ralph Stacey, who originally wanted to express with his work that people in organizations only become creative when neither the boundary conditions for achieving the goal are known nor there is agreement

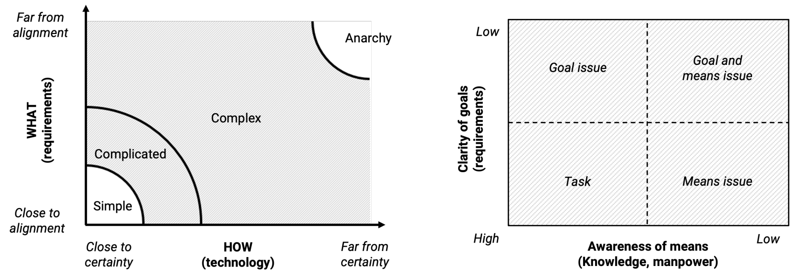


Figure 4.4
Stacey matrix based on Stacey (1996) and Zimmermann (2001), left; classification scheme for categorizing barrier types according to Dörner (1976), right.

on the actual goal (Stacey, 1996). In a current translation, this would be called moving out of the comfort zone. A detailed description of this can be found in (Atzberger et al., 2020a, p. 16). However, the crux is that he never created this matrix himself, but another scientist condensed his results into a matrix, which is completely correct in terms of the content (Zimmermann, 2001). This is per se not an issue, but this matrix has been simplified and adapted to agile approaches by other authors to such an extent that it serves as a classification scheme for agile methods. This simplified version is shown in figure 4.4 on the left side. The actual problem is that this representation lacks any scientific basis, and its content is simply too short-sighted, which is why it appears to be straightforward and easy to comprehend. Yet, the interconnecting links are missing, and thus the claim is invalid.

Another matter is referred to here as re-branding: In the right side of figure 4.4, the classification scheme for the classification of barrier types is displayed that distinguishes a construction task according to the means (knowledge, skills, material resources) as well as the goals (restrictions) in a matrix (Dörner, 1976). As one can see, the axis labels are very similar; only the content of the matrices differs marginally. From the perspective of a product development methodologist, the "agile" Stacey matrix is, therefore, a re-branding of an older, established model.

However, this is only one example out of many, as, e.g., the agile practice *Point estimate* or a *Burnup chart* are essentially not that novel, but rather slight variations of known techniques or instruments such as *pointing*[2] or a *project plan* (Lindemann, 2016, p. 296ff.). For this reason, the Taxonomy for agile development by Atzberger et al. has been developed to reconcile these "new" agile practices with the well-known and proven methods of classical design methodology to create clarity.

These results and the associated examples are intended to illustrate why the understanding of agile development is multifaceted. The resulting difficulties of interpretation will now be explained in more detail.

4.1.2 Interpretation issues

In addition to the ambiguity mentioned above, agility is often misinterpreted, which is reflected by an inadequate set of objectives. Since it is not precisely

[2] In German: Punkten.

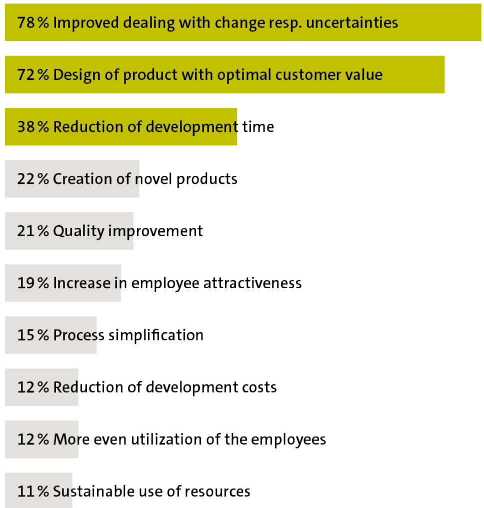


Figure 4.5
Aims of agile development (n = 106) according to Atzberger et al. (2020a).

specified what agility is and which goals agile ways of working are capable of pursuing, the goal definition is insufficient as well. This is elucidated in more detail in the following four aspects.

Goal-setting

The next dataset deals with the question of why agile development should be used, i.e., what goal is being pursued with its application (Atzberger et al., 2020a). For this purpose, the participants were given a list of ten aspects to choose from, whereby the participants had the opportunity to select several aspects and also add their own objectives. A total of 106 people answered this question. As a result, *improved responsiveness to change and uncertainty* and the *design of products with optimal customer value* were most frequently mentioned, representing the core of agile ways of working according to the original definitions, as displayed in figure 4.5.

However, the third most frequently cited reason was a *reduction in development time*. This is remarkable, as the previous years' results have shown that hardly any benefit had been achieved regarding the "controlling KPIs" (quality, cost, time) by applying agile development (Schmidt et al., 2018b, 2019). Schmidt et al. point out that positive effects must first propagate through several instances in means-end relationships before noticeable changes in the KPIs are discernible. However, it is not surprising that a shortened time-to-market is ranked third since it is fueled by the *performance capacity* and the *guru problem*, as explained below. Nevertheless, this leads to expectations not being met – due to an interpretation problem.

Performance capacity

As mentioned in the previous section *Flawsome comparisons*, the distinction between plan-driven and agile process models and their performance capacity is short-sighted to some extent. As an "advantage" of agile product development, figure 4.5 shows the development progress in classical and agile process models and differentiates between development time and degree of realization or user needs fulfillment. Through an incremental and iterative workflow, a reduction in development time and increased fulfillment of user needs are achieved (Schuh

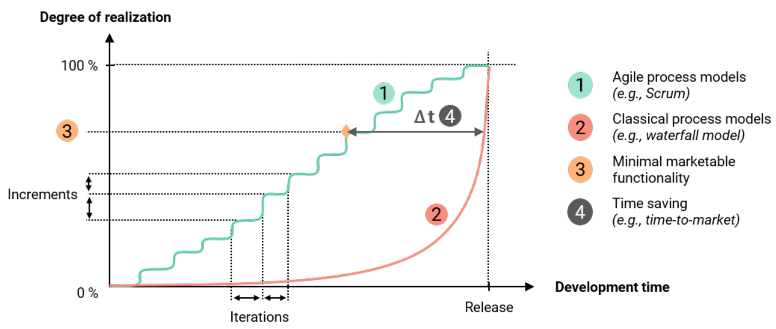


Figure 4.6
"Advantage" of a shortened time-to-market in agile development according to Schuh et al. (2017).

et al., 2017; Kantelberg, 2018). According to Schuh et al., the start of production (SOP) in an agile project is even earlier than in a project carried out in a plan-driven manner, with an even higher fulfillment of user needs as initially outlined. None of the experts in the first round of interviews could confirm a reduction in development time from their own practical experience. Interviewee I mentioned that *"people think they can develop faster or solve problems faster, which is not true at all. Agile working, however, makes problems more visible and intermediate results are achieved more quickly".* This faster problem discovery is seen as a *"huge benefit"* by interviewee B as well. Interviewee J mentions that there is no time gain, because *"the efforts are spent in other ways, into synthesis and analysis"*, and also *"agile rituals such as stand-ups, plannings, etc. require time, i.e., the communication effort is significantly higher and more frequent, which must not be underestimated"* according to interviewee G.

↯ Discrepancy

The guru problem also drives the assumption that agile working actually leads to a shortened development time. This term was first coined by Janes and Succi and refers to the issue that the *"founding fathers postulate their methods, but lack a thorough explanation on why they work"* (Schmidt, 2019, p. v). He refers to those people who are the co-creators of the Manifesto for Agile Software Development postulating their own agile methodologies (Scrum according to Ken Schwaber, XP according to Kent Beck, etc.). Janes and Succi describe it as follows: *"The guru is the person with wisdom – he or she knows what, when, and how things should be done to achieve the desired goal. It is the interest of the guru to hide the assumptions on which the rules are based, on which previous works he or she based his or her findings, how he or she verified that what is claimed really works. [...] The guru has no advantage in making his or her followers independent adults, otherwise his or her role as a guru would vanish."* Furthermore, they state that gurus use *"luring metaphors and fascinating analogies"*(Janes and Succi, 2012, p. 222).

Guru problem

The co-creator of Scrum, Jeff Sutherland, has even released a book titled "Scrum – The Art of Doing Twice the Work in Half Time" (Sutherland, 2014), which is a catchphrase most agile practitioners have heard of – yet an increase in efficiency of 400 % has not been reported by any of the experts in the interviews. Interviewee I mentions that Sutherland has done agility a disservice with this headline. *"Instead, you do not get more done in the same time, not even in half the time, but you get more important things done in that time"*, according to interviewee E. *"Nowadays, you*

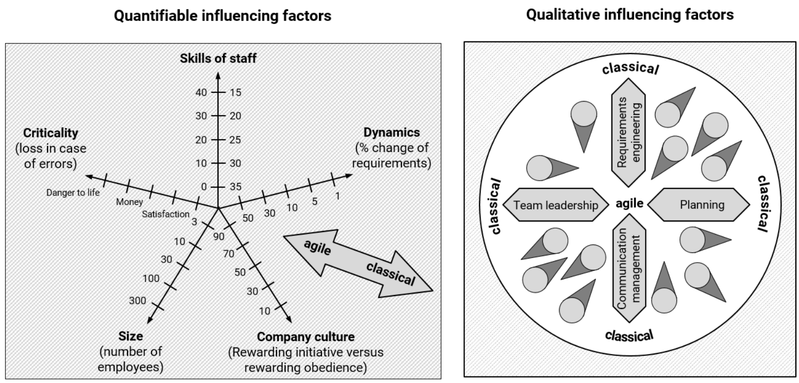


Figure 4.7
Criteria for the selection of a classical or agile process model according to Hruschka et al. (2009), left, and Welge and Friedrich (2012), right.

really have to be careful when using the word 'agile' because it is not well received everywhere, which is definitely due to the guru problem", notes interviewee C.

Selection criteria

There are selection criteria that serve as a decision-making basis for deciding whether it is appropriate to manage a project in an agile or a plan-driven way. Figure 4.7 shows two models that try to make the selection distinguishable based on quantifiable (left) and qualitative criteria (right). The model according to Hruschka et al. uses five quantifiable criteria, which are divided into the criticality of the project, personnel, dynamics concerning changing requirements, company culture, and the number of employees, with the sweet spot for agile working methods located in the middle. The second model according to Welge and Friedrich is of qualitative nature and consists of 13 individual aspects, which are categorized in team leadership, requirements engineering, planning, and communication management. A qualitative tendency is given for each aspect, and the ideal spot is located in the middle of the model. Hruschka et al. speak here of "standard territories" in which one type is superior to the other. However, the authors note that projects are in-between in reality, and these criteria are only tendencies. Welge and Friedrich highlight that a synergetic use of agile and plan-driven approaches is already perceived in software and can thus be transferred to other sectors. They recommend the targeted selection of specific agile process models or practices to transfer to the classical product development process.

Although these indicators are an initial support measure, the models have been developed for the context of the software. The context-specific adaptation is discussed in the following chapter. As a criterion for applying agile process models in the context of physical product development, the degree of innovation plays a decisive role. Albers et al. have shown that the degree of innovation strongly depends on the type of product and the sector, whereupon they have developed the concept of product generation development, which is mentioned in chapter 2.1.3. Therefore, in a different context, other contextual criteria need to be considered. In addition, various agile process models have strengths and weaknesses in different areas, which must also be considered in the evaluation. For this reason, the compasses are a first aid for orientation, but without a deeper understanding of how specific methods work, they lead to interpretation issues when being applied in an unsuitable context.

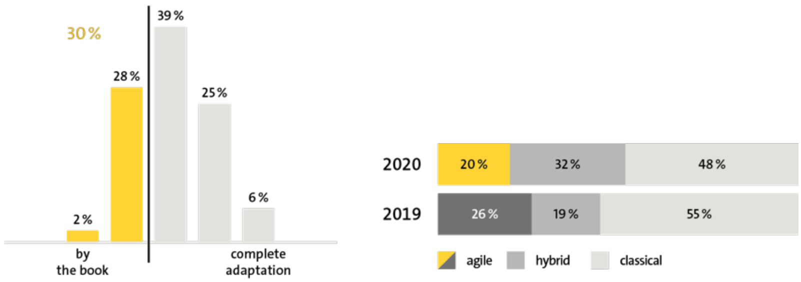


Figure 4.8
Degree of adaptation of agile process models (n = 112, left); percentage of all product development projects, broken down by type (2020: n = 99; 2019: n = 65) according to Atzberger et al. (2020a).

4.1.3 Implementation problems

In addition to difficulties in understanding and misinterpretation, problems regarding the implementation of agile working methods in mechatronics exist. In particular, this involves the transfer of agile process models that have proven themselves in software to the context of mechatronic systems.

High degree of adaptation

The status quo regarding the use of agile process models in physical product development is surveyed annually in the study by Atzberger et al. (2020a). For three years now, Scrum has been the front-runner (2020: 90 %), followed by Kanban (75 %) and Design Thinking (52 %).[3] However, it is noticeable that the longer companies deal with the subject, the more they modify and create their own procedural models. Hence, customized "own process models" was mentioned fourth (39 %) with a clear gap to the previous year (2019: 11 %). Participants were asked to rate how much they adapt their process models – from *by the book* to *complete adaptation* on a 5-point Likert scale. 112 participants voted and the results in figure 4.8, left side, show that only 30 % directly adopt or marginally adapt the chosen process model. Conversely, more than two-thirds feel the urge to adapt. This indicates that the process models that were once developed in the context of software cannot simply be used in the context of mechatronics without adaptations.

Furthermore, the participants were asked whether their product development projects are carried out in an agile, plan-driven, or hybrid form. The results from 2020 and 2019 are shown in figure 4.8, right side. On the one hand, it becomes evident that the share of plan-driven development projects is decreasing. On the other hand, it is noticeable that the number of projects carried out in an agile way decreases on average, yet the proportion of hybrid projects is increasing significantly. This trend is particularly due to the fact that there are context-specific differences between software development and mechatronic products that need to be addressed.

Context description

The agile process models have been developed in and for the context of software, as already mentioned. Adopting specific instruments and practices without context-specific adaptation and alignment will result in problems that affect the performance capacity. Therefore, it is necessary to have a precise understanding of the context into which a practice or an instrument shall be transferred so that it is known which connecting points must be taken into account and which levers can be used. Since the current context description between soft-

[3] Scaling frameworks were surveyed separately.

ware and hardware (mechatronics) is insufficient for agile process models, further differentiation criteria are necessary. As illustrated in figure 2.12 in chapter 2.3.1, software and hardware products differ in their appearance (virtual/physical) and the effort required to create and implement them in the different phases of the product life cycle. In this regard, it is necessary to further break down the distinguishing criteria.

Rigby et al. attempted to divide the *right conditions* for agile, i.e., the criteria for selecting an agile project when being applied in other domains like software, into favorable and unfavorable conditions. The criteria are market environment, customer involvement, innovation type, modularity of work, and impact of interim mistakes. Another rather simplified model represents the *sweet spot for agile hardware development* among the three dimensions team size, company size, and product architecture (Atzberger and Paetzold, 2019). Ambler and Lines provide an overview of potential factors to consider when choosing a way of working, which originates from the Disciplined Agile framework and has been adopted by the PMI (Project Management Institute). They are divided into selection and scaling factors, and the interaction of these twelve context factors and their influences are illustrated accordingly. In addition, a quantitative model (similar to the one in figure 4.8) has been come up with by PMI, which is divided into seven dimensions and already provides discrete levels per dimension as "tactical scaling factors". This type of compass can be used as a support, but it does not yet sufficiently address the context of mechatronics due to its generality.

The greatest obstacles to the successful application of agile working methods can be found on the technical side in the so-called constraints of physicality (Atzberger et al., 2020a). In the course of the studies on the current state of agile development in mechatronics, the handling of long-running processes due to waiting times in manufacturing and delivery of special components as well as the fulfillment of external requirements such as certifications were named as dependencies characteristic for physical product development (Schmidt et al., 2019; Atzberger et al., 2020a). Due to these frequent waiting times, multi-project management is also a central problem, especially in physical product development. Hence, a full-time dedication of team members to a project is easier to achieve in software due to the lack of waiting times (Cooper, 2016). These hindrances are inherent in the development of hardware and therefore not taken into account by the software as it does not occur there. For this reason, the context-specific characteristics must be taken into account when transferring methods from other contexts. Hence, identifying the differences in contexts is crucial for successful adaptation at the operational level.

Transfer of practices SW → HW

The direct transfer of practices from the software causes problems due to the neglect of the context, which is the result of difficulties in understanding. Interviewee P noted here that *"the transition from expert teams consisting of only one domain (e.g., mechanical or electrical engineers only) to feature teams that are staffed interdisciplinary has been quite challenging."* Interviewee R mentions that *"creating a PDI (potentially deliverable increment) every sprint in the context of plant engineering is simply not feasible according to the original definition."* Of course, detours can be found here if one does not adhere to this core element, as both Interviewee L and Interviewee R agree. *"The incremental approach is an essential part of agile working, but the* *increment serves as a medium to collect feedback. A potentially deliverable product every sprint is not feasible in hardware, and it does not have to be"*, mentions Interviewee L. This is where the guru problem comes up again because according to the Scrum guide, the guideline for the Scrum methodology, the methodology should be implemented precisely as described in the textbook; otherwise, it is no longer Scrum (Schwaber and Sutherland, 2017). At the same time, the guide states that Scrum can now also be used outside of software (Schwaber and Sutherland, 2017, 2020). Hence, such specifications inevitably lead to misunderstandings. Moreover, interviewee G states that *"a physical product cannot be cut arbitrarily due to the technical-physical dependencies, so prioritization is not expedient here, as physical laws must be obeyed."* As illustrated, some of these problems are mutually dependent, and therefore an approach is needed that not only solves the sub-problems but also deals with them differently.

4.2 Solution approach

In the previous chapters, the individual problem areas understanding, interpretation, and implementation were presented in detail. These areas are intertwined, as the implementation challenges are due to difficulties in interpretation, which in turn result from a lack of understanding. A central aspect considered here is that the "concept of agility" was developed in and for the field of software, and the mere transfer of agile process models entails problems, as previously indicated.

Context-appropriate transfer Since creating a virtual product differs considerably from a mechatronic product due to the varying boundary conditions, a description of the contextual differences is necessary. Based on these differences, a context-appropriate transfer can take place. In the following, three possible approaches for transferring agility from software to the context of hardware are shown using the St. Gallen management model, which is depicted in figure 4.9.

One possibility is to transfer at the operational level, i.e., to adopt concrete tools, practices, or methods directly, such as a retrospective or a daily stand-up meeting. However, only individual elements are selected and transferred to the other context. This approach can be a simple and effective way to learn how agile practices work if elements are selected that appear to make sense. However, dogmatically adopting an agile instrument such as user stories can quickly backfire if the purpose of the instrument is not understood. Approach 1

The second approach deals with adopting methodologies and process models in a new context on the strategic layer, such as the transfer of the Scrum methodology from software to hardware. As figure 4.8 shows, there is a need for adaptation, as the impact cannot be transferred across contexts without encountering problems. This is the most common way to transfer agile process models into a new context. Approach 2

The third approach is displayed as a grey arrow in figure 4.9, which anticipates an alignment of agile modes of action at the normative level. A two-sided arrow has been deliberately chosen here, as this approach represents an alignment of Approach 3

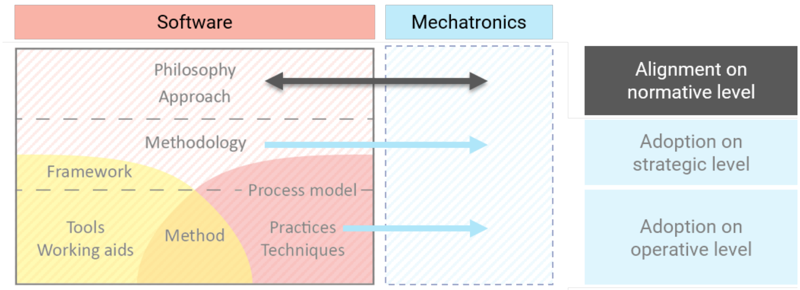


Figure 4.9
Approaches for the cross-contextual transfer of agile modes of action.

the contexts of software and mechatronics, in which a possible adoption of agile modes of action is to be achieved. On the one hand, this approach has the advantage that the elements at the normative level are universal in nature and therefore have cross-context validity, as Interviewee B and Interviewee L pointed out. On the other hand, they are thus generic and not directly applicable at the operational level, which needs to be taken into account.
This dissertation aims to align the contexts at the normative level to overcome the context-specific problems and thus develop a model that takes into account agile modes of action while considering the mechatronics context. The current efforts to unite agile and plan-driven approaches (cf. chapter 2.3.2) try to combine "the best of both worlds", but there is no consideration of the context here either, as agile process models at the strategic level are merged into plan-driven macro-level models. The model developed in this dissertation is intended to help grasp what characterizes agility in the context of mechatronics on a normative level. Once that is clarified, a subsequent breakdown into suitable, context-dependent elements is expedient.

4.3 Criteria for model development

This work aims to develop a model to describe the mechanisms of agile development in the context of mechatronics development. According to Patzak, a *"model represents a real state of affairs in an abstracting and thus simplifying way; it usually serves to predict the behavior of the real system"* (Patzak, 1982, p. 307). The purpose of a model is to be able to manipulate it as a representation of the actual, real system, whether for technical-physical, economic, or ethical reasons (Patzak, 1982). Requirements are placed on the development and validation of the model to ensure its validity. Therefore, the requirements that a model resulting from applied research needs to fulfill, according to the research process by Ulrich and Hill, are presented here. They are divided into content-related and formal requirements and are therefore an integral part of this work. These criteria support the target-oriented investigation of the most appropriate research area and serve to ensure the validity of the results. On this basis, the formal criteria are used to assess the validity of the model in chapter 6.

Content-related requirements The criteria regarding the content for the model development serve to structure and validate the contents of the model. The following four criteria are used here[4]: first, the *objective* of the work is crucial, which is to be achieved. Second, the *deficits in practice* and third, the *deficits in theory* are two additional criteria that represent the necessity of the work. Fourth, the *observation scope* is another criterion of interest that limits and classifies the work.

- The objective of this thesis (detailed in chapter 1) is about developing a model that shows the necessary properties that are required for agile development in mechatronics. For this purpose, the modes of action are examined at the normative level to identify all values that are essential for a cross-contextual use of agility. Objective
- The deficits in practice, which can be summarized under the term mindset, were elaborately explained in chapter 4.1. There are deficits in understanding the concept of agility, in the interpretation of certain practices and elements, and in the implementation of agile process models in the context of physical product development. Deficits in practice
- From a theoretical point of view, there is a paucity of publications dealing with agile development on a normative level, as outlined in chapter 2.3.3. Furthermore, there is no model available that describes agility in the context of mechatronics based on values and thus supports the alignment of agile modes of action across contexts instead of a systematic transfer of agile process models. Deficits in theory
- The scope of this study is defined in detail in chapter 2.4. It focuses on the agile development process in the context of mechatronics. Furthermore, the agile modes of action are analyzed on the normative level based on values. The scope of the study is limited to the German-speaking DACH region. Scope of observation

In addition to these content-related criteria, which set out the field of action for the work in this condensed outline, the formal requirements for the model to be developed are set out below.

Formal requirements The formal criteria serve to ensure the general validity of the model and thus complement the content requirements that are placed on the model to ensure its applicability and correctness (Kantelberg, 2018). For this purpose, Patzak has defined the following five criteria, which are used to ensure the formal correctness of the model design in this work: Empirical correctness, formal correctness, purposefulness, operability, and simplicity.
Kantelberg gives a condensed definition of every criterion:

- Empirical correctness ("similar") is achieved when correspondence between the model and the observations of the system in reality is created. A high degree of analogy between reality and the system structure or system elements must be reached to accomplish this. Empirical correctness
- Formal correctness ("exact") is fulfilled if the reproducibility of the model is guaranteed in the sense of repeatability and verifiability. Thence, the model must be formally well-characterized and constructed without contradictions. Formal correctness
- Purposefulness ("productive") is fulfilled if the model addresses specifically posed questions with answers that are appropriate in terms of both content and form. Purposefulness

[4] Kantelberg has used the same criteria for outlining the content-related criteria for agile product development in his thesis.

Operability

- Operability ("user-friendly") is fulfilled when the applicability of the model and the interpretability of the results are made as simple as possible for the user.

Simplicity

- Simplicity ("cheap") is fulfilled if the effort of both model development and application is reduced to the greatest possible extent.

For a detailed description of the individual criteria, see Patzak (1982). These criteria are used in chapter 6 again to assess the validity of the AiM model.

⚠ Remark

Patzak notes that *"obviously, the first three evaluation criteria compete with the remaining two; each model is therefore a compromise"* (Patzak, 1982, p. 310). A trade-off must be made here in order to find an optimum. In the DfV approach, this trade-off is referred to as "value conflicts" and is described in more detail in chapter 7.2. Furthermore, it also depends on the type of model and the purpose pursued with the model, which Patzak breaks down in more detail. The following description applies to the work: According to Ulrich and Hill, a real-scientific model can be defined as a "homomorphic representation" of a real system, i.e., as a structurally identical but simplified representation that abstracts from all properties that are considered unessential. In this context, Stachowiak has defined three characteristics every model maintains – a representative, a reductive, and a pragmatic characteristic. These properties of models need to be considered for further development.

In addition to the research procedure from chapter 3 and the problem identification as well as the solution approach to solving them, the following chapter shows the actual model conceptualization based on the criteria presented here.

5 Model conceptualization

This chapter describes how the different pieces of the puzzle intertwine with the development of the final model. *Chapter 5.1* discusses three different perspectives taken to examine the topic and presents which kind of data is used for the investigation. The following steps build on these databases, which is why their selection is crucial.

The combination of the DfV approach with the Taxonomy for agile development used in the analysis and further processing of the data is described in *chapter 5.2*.

Chapter 5.3 briefly summarizes the most relevant aspects in the course of the validation of the results. Therefore, the methods used for ensuring the validity of the results are outlined.

The final result, the **A**gility **i**n **M**echatronics model (AiM model in short) is presented in *chapter 5.4*. After describing the individual elements of the model, the results are constrasted based on the different perspectives initially taken.

Figure 5.1 illustrates the chronological relationship between the individual steps that have been taken in the course of this investigation. Starting from chapters 5.1 and 5.2, which are strongly concerned with the actual development, the focus changes to validation in chapter 5.3. This work aims to identify the essential elements resulting from three different perspectives initially chosen, as shown here.

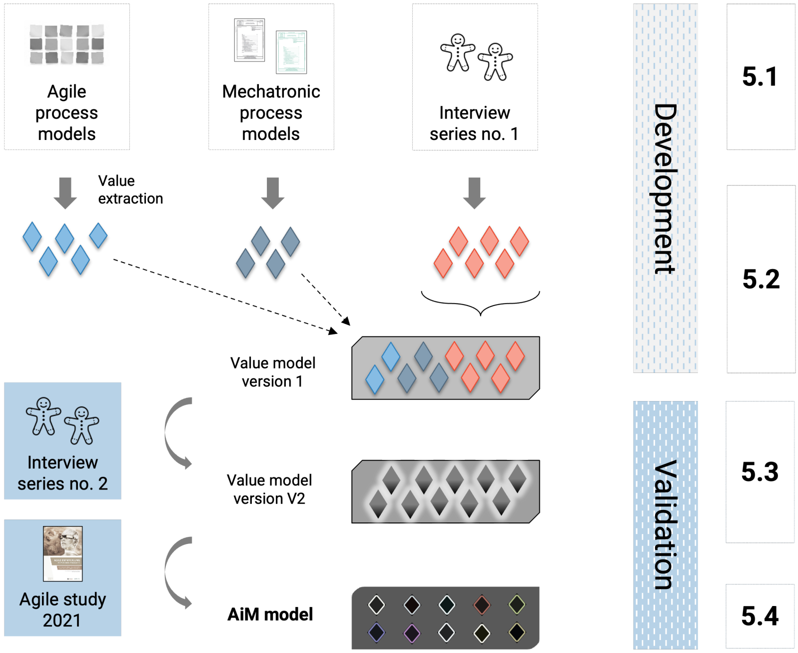


Figure 5.1
Chronological sequence of model development in this thesis.

5.1 Perspectives and data used

The goal of the initial step of developing the AiM model is to outline the different perspectives on the subject to comprehend which data have been used for the analysis. A *perspective* represents a *point of view* in the context of this work, or rather a *reference point*. From this reference point, the topic of interest is evaluated. Three perspectives were selected to explore the topic from different angles and determine in which aspects the results differ from these viewpoints. In this chapter, these perspectives are presented, and the respective input data chosen for the investigation is provided, as displayed in figure 5.2. The three perspectives are as follows:

Perspective 1
Agile process models in software development

The first reference point taken is the view from agile software development. Since agile methodologies have extensively been used in the software domain, many experiences regarding the characteristics of an agile process have been gathered over time. Here, the perspective of an agile practitioner is taken.

Perspective 2
Methodologies in mechatronics development

As a second reference point, the view from classical (plan-driven) mechatronics development is taken. Product development (as a field of study) has a long history, dating back more than a century and a half, as mentioned in the state of the art, and therefore many achievements have already come up in this field. Accordingly, the general conditions for which these classical (plan-driven) process models were developed have been investigated thoroughly. As a result, the prevailing methodologies (as described in chapter 2.1.2) have been adapted and aligned to these contexts.

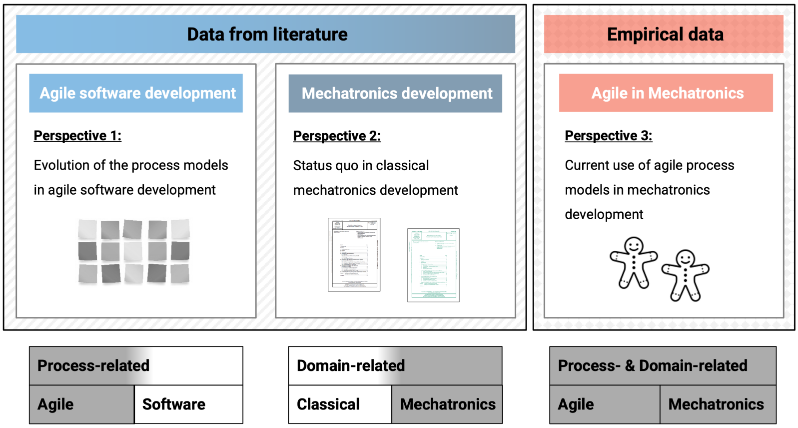


Figure 5.2
The three perspectives used for this investigation.

Perspective 3
Agility in the development of mechatronic products

The third reference point is the actual perspective of study, agility in mechatronics development. Based on the existing experience of practitioners from the industry with the application of agile methodologies in developing physical products, this perspective is discussed and further elaborated upon. As this field is still relatively new, the other two perspectives will be used to highlight the exact differences as well as commonalities between them. This is carried out in chapter 5.4 after the description of the final model.

The perspectives are explained below in more detail. These are each subdivided into their aim and the corresponding content. The data which has been selected for the investigation is presented. Based on the selection, the actual analysis is carried out in the next chapter.

Aim

Perspective 1: Agile process models in software development In chapter 2.2.3, various process models were presented in the context of agile software development, which can be divided into the *agile software period* and the *scaled agile period*. These two periods will also be adhered to for the following analysis. The aim is to obtain a comprehensive snapshot of all relevant process models of agile software development in these two periods to be able to extract the corresponding elements in the next step (for these two periods) and to present the evolution of the process models in the context of software development.

Content

Overall, all process models that were recorded in the chronicle were considered for the study. Of the 24 possible process models, *Kanban*, *Design Thinking*, and *Lean* were excluded because they were not originally developed in the context of software, as shown in chapter 2.2.3. Furthermore, *SoS* was not considered, as it is only a minor extension of Scrum and does not provide any further insights. The hybrids *ScrumXP* and *Scrumban* were not considered either, as they are already considered in their purest form. Their combination does not yield any new insights (at the normative level), as they do not introduce any new aspects. Other hybrid approaches (as pointed out by Heimicke et al.) are not included, as they do not provide any further insights for the following analysis either (as they are merely a combination of existing elements, too). In total, 18 process models of agile software development are included in the analysis, as summarized in table

Table 5.1
Overview of process models used for the investigation.

Agile software period	Scaled agile period
RAD	LeSS
Scrum	DevOps
DSDM	SAFe
Crystal	DAD
XP	Spotify
FDD	Nexus
ASD	Scrum@Scale
TDD	
AM	
DDD	
BDD	
Σ **11**	Σ **7**

5.1. These process models can be assigned to either of the two periods mentioned above and are or have been used to a certain extent and are therefore relevant for evaluation.

Aim

Perspective 2: Methodologies in mechatronics development In this perspective, the current process models of mechatronic system development are presented. Here, the antithesis of what is referred to as "agile" will be presented in its most current form. This aspect is of great importance, as in several other publications on agile development, the reference point set (of a plan-driven approach) does not correspond to the current happening, as outlined in chapter 4.1.1 (cf. *flawsome comparisons*). Hence, this perspective aims to represent the approaches used in the classical (plan-driven) development of mechatronic products in their most current form.

Content

For answering the research question at hand, topical process models are of relevance since it concerns the context of mechatronics development. Thus, only current process models (2010 and later) have been taken into account for the investigation. Hence, chapter 2.1.2 describes a total of seven methodologies and models in the context of mechatronics development. However, as far as the scope of the second perspective is concerned, only two process models are used for the evaluation, representing the current form of "classical" mechatronics development.

The four-cycle model is, by name, merely a model, not a methodology. Although Pahl/Beitz is a standard work in the education of engineers and has recently been updated, it does not represent a process model per se. Therefore, these two are not included in the analysis. The MVPE focuses on the virtual support of the product life cycle and does not provide any normative elements, so this is not considered either. The W model is merely an extension of the earlier version of the V model (2004) and therefore does not provide any new elements. The iPeM has been incorporated into the revised version of VDI 2221, sheet 2 (2020) and is thus comprehensively represented.

In conclusion, VDI 2206 and VDI 2221 are considered as the current reference points in this perspective because of their currency and context-specific use. Even the old version of the VDI 2221 from 1993 is unsuitable because the context has changed completely, and is thus not up-to-date.

The following should be noted here: The study of design methodology[1] has a long history, particularly in Germany, dating back over 160 years. Some authors have already partially mapped the historical account of the development of methodologies in the German-speaking region, displaying how and when particular movements have started or evolved at certain points in history. The contextual embedding of the methodology in its regional context is of particular importance due to many achievements and contributions to design methodology by authors from the DACH region. Although the experiences have always been incorporated into the new procedural models, for the investigations in the context of this work, in addition to the cultural fit, topicality is of utmost importance.

⚠ Remark

Perspective 3: Agility in mechatronics development The goal of this perspective is to illustrate what characterizes agile development in mechatronics. This perspective represents the blending of the two previously mentioned perspectives, as highlighted in the bottom part of figure 5.2 (grey). However, as this perspective is relatively new and thus under-researched, the results in this perspective come from empirical research. In conclusion, the results are compared to the other two perspectives in chapter 5.4 by using the AiM model.

Aim

A total of 13 expert interviews were conducted for this purpose as part of the analysis. Except for the last person, the interviewees are part of the VDI expert committee VDI-GPP 710 "Agile Development of Mechatronic Systems". Details about the interviewees can be found in table 3.1 or appendix A.6. Ten people are from industry, two people are university employees, and one person is a consultant. All of these people have several years of experience with the topic, which is why they have been selected to be part of the committee by the VDI. The participants have an average of 5.6 years of experience with the topic. They received interview information in advance, and the interviews were conducted in a semi-structured way. The interviews took place in September 2020 and lasted 42 min on average. Due to safety protocols (because of the Covid-19 pandemic), all interviews took place remotely, as outlined in chapter 3.3.2.

Content

Based on the expert interviews, the results of this interview series №1 thus represent the fundamental understanding of what characterizes agility in mechatronics.

5.2 Data analysis

In the following chapter, the three data sets from the perspectives are analyzed in more detail. For this purpose, the alignment approach which has been chosen is elaborated upon first. After that, the actual procedure of the analysis is

[1] In German, the term *design* can relate to "Entwicklung" (development) or "Konstruktion" (to construct or design something), therefore in German one can distinguish between ‘Konstruktionsmethodik’, as by Hansen and ‘Entwicklungsmethodik’, as in VDI 2221.

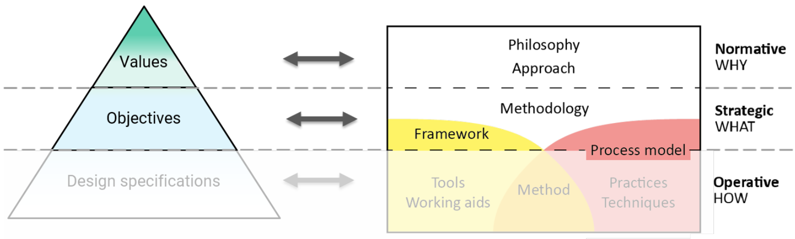


Figure 5.3
Assimilation of values hierarchy method (left) and Taxonomy for agile development (right).

explained, and an intermediate result of one perspective is presented as an example.

Presentation of the approach taken

For the analysis of the perspectives, two models have been merged. On the one hand, the *values hierarchy method* representing a central component of the Design for Values approach, as described in chapter 3.2, was chosen. On the other hand, the *Taxonomy for agile development*, as described in chapter 2.2.2, was used. These two models have been assimilated in order to be comparable. Both models exhibit the commonality that they are divided into three levels: The values hierarchy method divides into the values, objectives, and design specifications level. The taxonomy, which is based on the structure of the St. Gallen management model, is divided into the normative, strategic, and operational level. Both models are displayed in figure 5.3, and dashed lines indicate their assimilation.[2]
As process models are located at the strategic level, it is difficult to compare them at that level due to their contextual dependency. However, to make the different process models from both agile and classical development comparable, they can be examined based on their underlying values. If the bottom-up approach is applied and the values are extracted from the different process models, it is possible to reconcile the modes of action of agile and classical methodologies on the normative layer.
Figure 5.4 is intended to illustrate the procedure of element extraction and aggregation as a Venn diagram. It is important to note that some elements are inherent in several methodologies, as illustrated by the overlapping circles. Hence, both similarities and differences between the methodologies can be illustrated by this visualization. This work aims to determine the normative elements that characterize agility in mechatronics development, which is visualized by the elements within the intersection in figure 5.4. For the sake of completeness, an additional circle with the values extracted from empirical research (interview series №1) is added in this illustration. The intermediate result in the course of the value extraction is a large list of all extracted normative elements for each of the respective perspectives.

Analysis regarding normative elements

First, the respective methodologies of the datasets outlined in the previous chapter were explicitly analyzed for their normative elements (using the sources provided). The terms *value* and *principle*, which are known as normative elements in agile software development (cf. Manifesto), were selected as keywords. As a result, all normative elements were extracted for each perspective and summa-

[2] The assimilation was verified by the authors of these two models to ensure their validity. Due to the similarity of the levels, the assimilation is therefore acceptable and substantially correct.

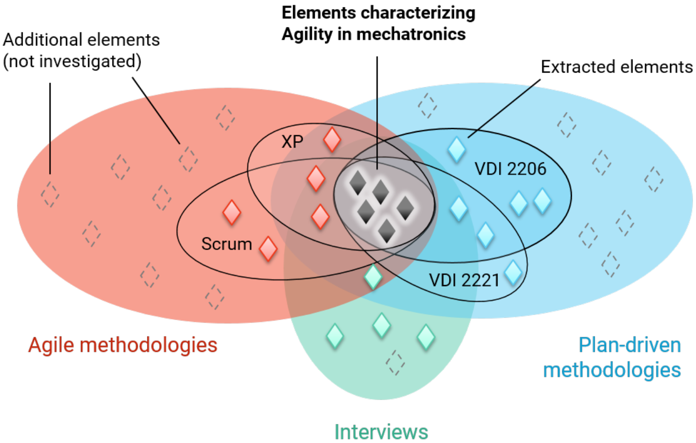


Figure 5.4
Extraction of normative elements from the methodologies and interviews.

rized in an overview, which is displayed in appendix A.3. In the next step, the normative elements for each perspective have been clustered according to their occurrence. Similar elements have been consolidated, and certain attributions have been aligned to the value level (e.g., in SAFe: Incremental build with fast learning cycles ⟶ *short-cyclic gain in insight*).
Based on this revision, the most frequent elements for the respective perspectives have been identified. The most frequent mentions (at least two mentions after the revision) were used to characterize the respective perspective. Each normative element has been assigned to one of the three cornerstones[3] *product*, *process*, and *person*, depending on the contextual description in the sources. The outcomes of these mappings are presented in the intermediate results.

Intermediate results The intermediate result for the agile software period of perspective 1 is displayed in figure 5.5 representatively. Since this represents a work-in-progress state, the elements are arranged along a triangle, with the corners being labeled according to the aforementioned cornerstones product, process, and person. Utilizing this representation, the individual elements can be positioned along the spanned triangle. This is of particular interest since several terms (and their understanding) have different connotations, and this visualization enables to display the differences, especially between the triangles, for further analysis[4]. Moreover, the most frequently mentioned elements are highlighted in bold to reduce the focus of a perspective to only a few elements. In summary, the most important elements are shown in the intermediate results, which were used to develop the final model. The detailed representations for all three perspectives can be found in appendix A.4.

[3] The cornerstones follow the context description of Earl et al. (2005), when referring to the environment of product development Bender and Gericke (2021). This segmentation was used because the focus is on investigating the (agile) development process, however there are also elements that contribute to the other two cornerstones. In addition, the process can be depicted as the interaction of people towards a specific goal (product).

[4] This can best be illustrated by the example of simplicity. In the agile software period, the focus is on keeping both the product and the corresponding process simple. In the subsequent period, the understanding of simplicity has changed significantly in the direction of the product and away from the process. This is explained in more detail in chapter 5.4.2.

Remarks on the value extraction

One major hurdle in the value extraction process is the connotation of terms across contexts, meaning that the understanding of what is referred to as a value differs greatly between the different perspectives. For this reason, the term *normative element* is used in the following to avoid conceptual contradictions.

In perspective 1, the agile methodologies refer to *values* as well as *principles* as normative elements, as stated above. Principles in this context represent the concretization of values. However, the terms are not used without overlap, i.e., a certain aspect (e.g., feedback) is referred to as a value (XP, AM) or a principle (Crystal, ASD). Moreover, some methodologies only name certain values, others name exclusively principles and some both.

Neither values nor principles are explicitly listed in perspective 2 of the classical procedure models. Especially the term *principle* is partially occupied in German product development literature, and the term *value* has different connotations in German as well compared to English scientific literature. Therefore, the data was studied in-depth, and the aspects that most closely resemble a normative element were elicited and listed.

In perspective 3, the interviewees were facing difficulties in answering what they perceive as important values for agile development in the context of mechatronics, with the German denotation of a value in mind. When they were asked which objectives concerning the benefits they have experienced, the interviewees were able to name specific elements straightforward. Several important aspects were mentioned here, which is why they have all been compiled as well. The second layer of the values hierarchy method is referred to as *evaluation criteria* in another representation of the DfV approach (van den Hoven et al., 2015). This is very similar to these aspects since the (realized) benefit correlates with the objective, which is based on prescribed criteria for action. Using this compromise, it was then possible to determine the underlying elements by using the bottom-up approach. However, in the final presentation, all terms have been aligned to the normative level to ensure comparability.

⚠ Definition

A **normative element** is, therefore, a term that is reflected in the form of a *value* or *principle* or *mode of action* that can be located at the normative level of the St. Gallen management model. It features characteristics that are in line with the definition of final or instrumental values, as pointed out in chapter 3.2.

Based on these intermediate results, the findings were aggregated, and a first (preliminary) version of the AiM model was created. This further development and partial validation of the intermediate results are explained in more detail in the following chapter. Chapter 5.4 features the final AiM model comprising its ten normative elements.

5.3 Validation steps

First, the vehicle used to described the interrelations is described, and subsequently, the two validation steps are elucidated which were used to validate the model. The preliminary results from the previous step were discussed in more depth with the second round of experts and supplemented using the means-end

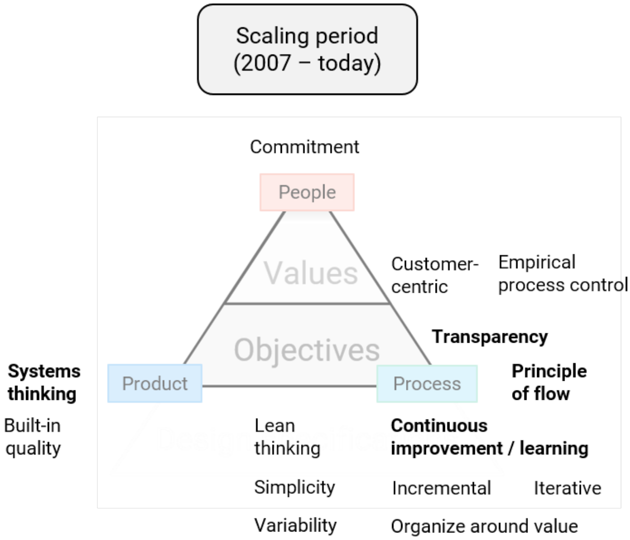


Figure 5.5
Schematic representation of an intermediate result of element extraction of perspective 1.

relationships. In a second step, these results were verified in a widespread and renowned survey in the DACH region with 99 participants. This quantitative investigation thus supports the qualitative data. The embedded design approach was chosen so that the essential elements could be identified using the input of selected experts in a first step. Following that, the community of practitioners was able to confirm these in a second step. This approach appeared to be most appropriate in the course of the investigations due to the issue of lacking clarity in terms of understanding, as described in chapter 4.1.

"Vehicle" means-end relationships

From the first set of interviews in perspective 3, the essential values and principles were extracted to understand how the different aspects are intertwined. For this purpose, it was analyzed which elements of the other two perspectives were also highly relevant, and these were added accordingly. The relationships were then sketched based on the transcribed conversations, and thus, the individual elements were linked. As a result, a comprehensive representation of interconnections has emerged, aiming to understand how the individual elements mutually interact.

In the end, means-end relationships emerged, which were discussed and adapted based on the comments. An extract of the means-end relationships is shown in figure 5.6. The detailed presentation can be found in appendix A.5. This "vehicle" was used to discuss with the experts of the second series of interviews how the individual elements interact with each other and, ultimately, which of these elements are fundamental. This type of representation shows how one certain aspect propagates through several instances and is therefore well suited to depict and thus discuss the reciprocal effects between the individual elements. This mapping is much more detailed than the description of Schmidt et al. (2018a), consequently. The focus of these means-end relationships is the completeness of the results and the correctness of the content so that the interrelationships are coherent.

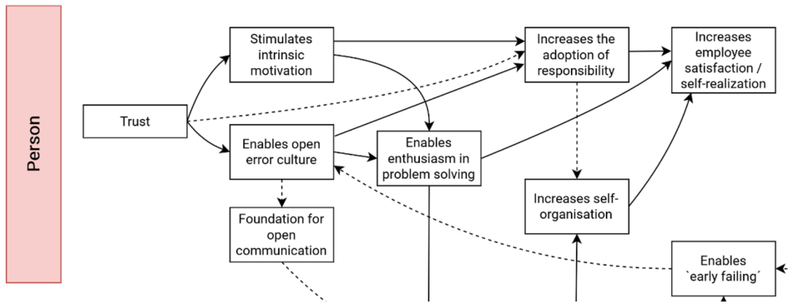


Figure 5.6
Excerpt from the means-end relationships.

These means-end relationships represent an aid to answer the research question and thus explain why (and especially how) agile development works, which has been called an unresolved issue by Schmidt to date. Moreover, the fundamental aspects form the elements of the AiM model, which are the basic constituents for a successful application.

Procedure

Qualitative step: Interview series №2 The first step to validate the results was conducted through the second series of interviews with experts from industry. This additional qualitative procedure was carried out after the consolidation of the perspectives to reflect the results with a second set of experts and thus fine-tune the results. First, the entire approach of the present work was presented to the experts to obtain feedback on the procedure itself. In the second step, the intermediate results and the means-end relationships were discussed with the experts. Based on the discussions, the interrelations within the means-end relationships were modified so that the basic elements could be identified in a final step. The selection of elements for the AiM model was carried out based on the discussion results on the means-end relationships. This represents the last step in the process of the element selection, and the results are presented in the following model description section.

Participants

A total of 7 experts took part in the second interview series. The first participant, interviewee L, was the only one to participate in the first interview series, too, since the participant has an extremely high level of experience with the topic. The remaining six interviewees are members of the APPe network (cf. chapter 3.3.2). This network is an association of companies in Germany that has come into existence in 2016 and comprises 16 companies and two universities. The participating companies in this network are pioneers in applying agile process models in physical product development, and therefore, the network is a closed circle. The participants from this group have many years of experience with the topic – 7.4 years on average, which is why they can critically assess the results as they are experts with the highest level of experience in the DACH region.

Procedure

Quantitative step: Validation by survey In the second step of the validation, the goal was to validate the results collected by the qualitative approach by a broad field of users. Since an empirical study on the topic of agile development of physical products is conducted annually at the author's institute, the findings of the previous validation step were queried in this context as well. The detailed approach is outlined in chapter 3.3.2. This procedure aims to validate the findings by many practitioners in the area of interest, the DACH region. Therefore, this confirmatory approach is seen as most appropriate to conclude within the overall investigation.

Participants

A total of 127 people from the DACH region took part in the survey.[5] A wide range of industries are represented, with half of the participants working in mechanical and plant engineering as well as vehicle and transport technology. All company sizes are well balanced, from SMEs[6] to large corporations. About half of the survey participants have staff responsibility and about one third are internal or external (agile) coaches. Almost 80 % of the participants are directly employed in development or pre-development departments. The majority of participants already have 3-5 years of experience with agile development in mechatronics, with an average of more than 5 years. The participants also have a high level of personal experience, with more than 85 % currently actively involved in agile development projects and therefore reporting from actual first-hand experience. On average, the participants' companies have been involved in agile mechatronics development themselves for about 3-5 years.

5.4 Agility in Mechatronics

In the following section, the final AiM model is presented, and its elements are explained in detail. Next, the results of the AiM model are compared to the findings of the two remaining perspectives, which have been elaborated in chapter 5.1. By contrasting the different perspectives, similarities and differences between the perspectives can be pointed out by canvassing the different characteristics of one element from varying perspectives. This enables the reader to comprehend how a certain element is connoted in this context and aims at providing a holistic and reflected understanding of which facets agility in the realm of modern product development bears.

5.4.1 The AiM model explained

The AiM model is intended to summarize the central normative elements that characterize agile development in the context of mechatronics. It is displayed in the shape of a tetrahedron instead of a triangle (as it has been further developed in the course of the validation). The corners of the tetrahedron base display the three cornerstones of the original triangle of chapter 5.2, being person, product, and process. The remaining corner, the tip of the tetrahedron, is intended to represent the customer, which follows the context description of Earl et al. (2005) in its entirety.

Since the descriptions of the corners person, product, process, and customer represent *cornerstones* within a product development process, they are referred to as such in the following. All of the central elements of the model are placed

[5] It was ensured that the participants actually develop mechatronic products and do not report from their experience with agile software development.

[6] Small and medium-sized enterprises.

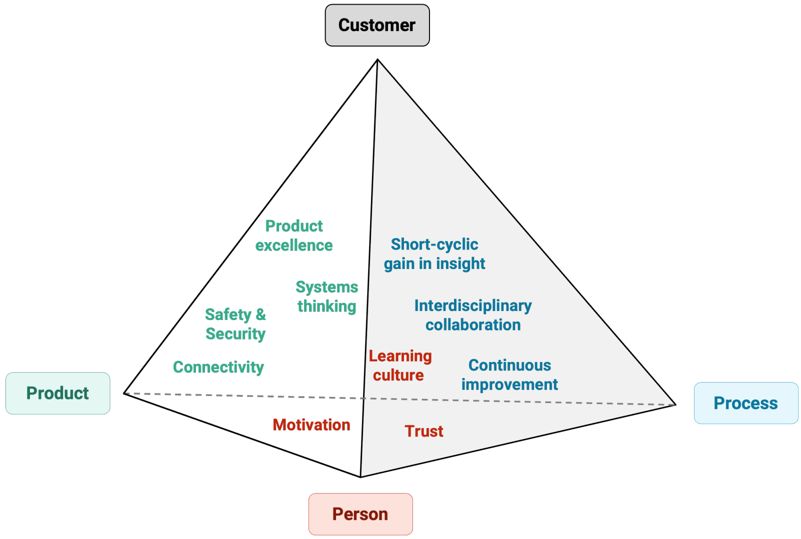


Figure 5.7
The AiM model and its normative elements.

within the tetrahedron at different coordinates.[7] This shall represent the relation of the element to the respective cornerstones, which differs between the elements. Nevertheless, each element is rooted at one cornerstone to which it has the strongest connection. The model is displayed in figure 5.7. For the sake of clarity and readability, a lateral view of the tetrahedron is chosen.

The AiM model displays the most relevant elements that have been identified in the course of the investigation. It consists of ten normative elements overall, which are embedded in the tetrahedron in colored letters. The values in the model are described below:

Element 1
Trust

From the human-centered perspective, one central element of agility is trust. According to the VSD scholars, trust is a human value that *"refers to expectations that exist between people who can experience goodwill, extend goodwill towards others, feel vulnerable, and experience betrayal"* (Delft Design for Values Institute, 2019). In the context of this model, the definition describes the connotation of trust very well. Here, trust does not refer to characteristics of the process or the product (i.e., relying on the process or product), but the interaction of the individuals within the context of professional work. As organizational theorist Stacey once described a company being "a body of people pursuing a certain goal" (Stacey 1996), the trust of one individual towards another person within this social but formal context is what is referred to in this model. Trust between these individuals is one of the fundamental requirements for agile development to be able to exploit its potentials.

Element 2
Learning culture

One aspect, which is in close accordance with the value of trust, is the element of learning culture. This is often referred to as *open error culture* and describes how a body of people perceives the act of having made a mistake. It is the perception (of the individuals around oneself) of the act of having failed which is of utmost

[7] The coordinates of the individual elements are displayed in table 5.2.

importance in this regard. This societal group of people can either condemn this failure and blame the person in charge of it or accept the wrongdoing and perceive it as an increase in insight. When having analyzed why and how the initial approach has not succeeded, this insight turns into an additional knowledge gain, which, in turn, stresses the act of learning. In this light, the startup scene idiom *fail fast, fail early* incorporates what the actual benefit of this element is – to mitigate the risk of pursuing a direction that will not be of value to the actual result. Of course, trust is a central precondition for a learning culture to flourish, yet it is not simply deductive. That is why it plays a central role in the AiM model and is one of the most ponderous elements.

Element 3
Motivation

Another human-centered element is the motivation of the individual. Motivation, in this regard, refers to the intrinsic pursuit of accomplishing a purposeful achievement, which is often related to enthusiasm as well. While this aspect is prevalently taken for granted, especially in an area where individuals are bound to long for creativity, it is a decisive factor in the process of design and development. As interviewee P has stated, "*motivation cannot be increased because it is an intrinsic characteristic of any human being. Thus, a person's motivation can only be diminished*" (by the surrounding context), yet it can be unlocked again as well. By providing a context in which several individuals can unlock their motivation, this motivation develops its "own dynamics," as interviewee A has referred to it. "*This bubbling and sparkling of ideas feels like constantly sitting on a witch's cauldron when the development is acted out in a purely agile manner. There is energy in it that can not be put into words"* (Interviewee A). However, such a high degree of own dynamics is only achieved when both trust and a learning (and thus failing) culture are put in place. From the human-centered perspective, all three elements are closely related, with trust at its basis.

Element 4
Interdisciplinary collaboration

One element describing the interaction of individuals from a process perspective is interdisciplinary collaboration. This explicitly refers to the interaction between individuals who come from different contexts yet pursue a common goal. Collaboration here is an extended and closer form of interaction compared to the mere cooperation of individuals. Here, a development team is not only staffed with people from the same domain, but these development teams are cross-functional – with people from different domains in the same team. By matching individuals of different backgrounds and expertise, the domain-specific performance shrinks, but the domain-overarching performance of the team rises due to its manifold expertise. The staffing of development teams from expertise to cross-functional teams requires changes in the organizational structure, as explicated in chapter 7.4 using a practical example. As referred to at the beginning of this dissertation, Melvin Conway once formulated what became known as Conway's Law: *"Organizations which design systems [..] are constrained to produce designs which are copies of the communication structures of these organizations"* (Conway, 1968). Taking the maxim 50 years to the present, communication structures could be substituted by team composition. Thus, by changing the structure of the (development) team, the results become apparent in changing communication structure and, ultimately, is directly represented in the product structure.

Element 5
Short-cyclic gain in insight

Another key element of the process is the short-cyclic gain in insight, which stresses the character of the process to be both incremental and iterative. The process is repeated in short cycles and carried out multiple times, which characterizes its iterative nature. Additionally, the process is incremental, meaning that it aims to produce an increment at the end of the iteration. In the context of mechatronics development, the so-called constraints of physicality hinder the production of a physical increment tremendously (Ovesen, 2012). However, the perception of what characterizes an increment needs to be adapted to the context of physical product development. For the process to still be incremental, *pirtypes* are also seen as valid increments, which do not necessarily have to be of physical nature (Schmidt, 2019). The intention is that any type of increment that adds value by generating insights about the product to be developed is perceived as beneficial and thus extend the current state of knowledge. By constantly generating increments in predefined time intervals, which are then validated, the learning curve of the development team increases steadily.

Element 6
Continuous improvement

In line with the element mentioned above, continuous improvement focuses on learning within the process and constant feedback of information. While the idea of continuously improving the status quo has its roots in *Kaizen*, it is an important aspect within any cyclic process to date. The continuous feedback of information about any specific detail within the process is intended to enhance the outcome overall, on the product, process, and person side. This additional feedback loop (the continuous improvement process) represents a process in itself. By incorporating the lessons learned into the initial setting, the status quo is gradually but constantly improved, and thus, the actual outcome tends to become more robust. Additionally, changes or hindrances can be addressed more easily since slight variations are the standard, not the exception. Hence, by applying measures such as e.g., a retrospective, the process can readjust and respond to them. In addition to both human-centered and process-related aspects, elements that need to be incorporated in the development of the actual product (or system) as the outcome of every development process need to be considered. On the product side, it is important to mention that only those elements are listed, which are of interest across (almost) all branches. Aspects, which are relevant only to specific branches, are therefore not taken into account since these are not generally valid. This has already been discussed in greater detail in chapter 3.2.

Element 7
Systems thinking

From the product point of view, systems thinking is one central element. This term is coined by Russell Ackhoff, who relates to it as follows: *"System thinking is holistic; it attempts to derive [an] understanding of parts from the behavior and properties of wholes, rather than derive the behavior and properties of wholes from those of their parts"* (Ackoff, 1999). Moreover, "*because a system is a whole that cannot be divided into independent parts, its performance is never equal to the sum of the actions of its parts taken separately; it is a function of the interactions. It can be shown that when each part of a system taken separately is made to perform as well as possible, the system as a whole cannot perform as well as possible*" (Ackoff, 1994). While having played an important role in systems engineering over the past decades, the rise in complexity of individual systems has increased significantly to date as well. Interviewee B has given a vivid example of this with a customer-focused approach: *"We notice* *that silo thinking provides us with highly optimized products, which is great, and sells well. However, our customers increasingly want us to provide them with a complete package so that the machines are inter-connected and the result is as good as possible. [...] The goal is, therefore, to find an optimum for this overall system. That is why it might make sense that one machine is two milliseconds slower at one point so that we can get five seconds out of the whole process. This kind of thinking does not work in a classical hierarchical organization, in which a person is measured by how good his/her machine performs, but only in an organization, which shows that it functions reasonably and thus finds its optimum"* (Interviewee B).

Element 8
Connectivity

This example has already picked up on the following element, which is connectivity. In this context, connectivity refers to the ability of systems to be interconnected, which is an aspect that is technology-driven and state of the art in the development of cyber-physical products nowadays. Connectivity as a central element in the product characteristic has different facets, as it relates to the networking of different systems to generate a superordinate system and the linking of different individuals with one another. In this context, these can be two or more people who already know each other and are more closely connected by the product or system, or it could be a customer-manufacturer relationship. Besides the actual technical realization of the product, this element can have a large impact on the product's success when taking a look at, e.g., the influence of social media in today's society. The ability to access information due to apparent and hidden data flows enhances the product's performance significantly from a customer's point of view. An example could be the real-time display/data transmission of available charging stations in a Tesla automobile, calculating the most time-efficient route to reach the desired destination. This element has also been given special emphasis in the revised version of VDI 2221 (2019a) since this is one distinguishing feature compared to the past.

Element 9
Safety & Security

Next, there is an element which is ambiguous in its German linguistic usage. Since the investigation has been carried out in the German-speaking region, the German term *Sicherheit* implies both the meaning of safety and security. While definitions tend to differentiate between security being mostly an external aspect while safety is an internal affair, one can distinguish that security protects against deliberate threats, whereas safety refers to protecting against unintended threats (Line et al., 2006; HSE, 2018). In the context of mechatronics, safety refers to securing the system and its subsystems against a malfunction that could either reduce the system's operability or (in the worst case) even harm the individual. In contrast, security represents a deliberate countermeasure one takes against a possible yet intended threat. Cybersecurity is a vivid example that aims at protecting any kind of system against intended external intrusion (Nord, 2016). For the model at hand, both meanings are specifically included since they are within the German definition.

Element 10
Product excellence

The final element of the model is referred to as product excellence. The facets of this term are twofold: *first*, the product being developed should be the best possible solution available. In other words, the product should solve the task in the best possible way, i.e., most efficiently. This is what the German industry has stood for over decades, delivering high-quality products through sophisti-

Table 5.2
Influence of elements on each of the cornerstones.

№	Element	Person	Process	Product	Customer
1	Trust	●●●	●○○	●○○	●○○
2	Motivation	●●●	●○○	●○○	●○○
3	Learning culture	●●●	●●●	●●●	●○○
4	Interdisciplinary collaboration	●●●	●●●	●○○	●●○
5	Continuous improvement	●●○	●●●	●●○	●○○
6	Short-cyclic gain in insight	●●○	●●●	●●●	●●○
7	Systems thinking	●○○	●●○	●●●	●●○
8	Connectivity	●○○	●○○	●●●	●●○
9	Safety & Security	●○○	●○○	●●●	●●○
10	Product excellence	●○○	●●○	●●●	●●●

●○○ weak relation
●●○ medium relation
●●● strong relation

cated engineering technology. The *second* facet is that the product should solve the customer's need in the best possible manner, i.e., it should solve exactly the customer's problem. However, the strong customer focus is a more recent aspect that is increasingly receiving attention across all industries since it ensures the desirability of the product for the intended customer segments. The maxim behind the element "product excellence" is reached when matching these two aspects.

As already mentioned at the beginning, the cornerstones person, product, and process form the basis of this tetrahedron model. The tip of the tetrahedron should represent the customer because the overall aim of an agile development process should be focused on delivering value to the customers by focusing on their needs. Due to the two-dimensional illustration of the AiM model, it is not distinct to what extent a particular element is related to all four dimensions. For this reason, table 5.2 shows the interrelation between the individual elements and each cornerstone in the form of dots. Here, ●○○ displays a rather weak relation of the element to the cornerstone, ●●○ a medium relation, and ●●● a very strong relation. The classification has been done based on the discussion with the experts in the validation step in chapter 5.3. This table is intended to display the interrelations of the elements to all four cornerstones comprehensively so that an actual three-dimensional representation could be produced.

⚠ Remark:
Dots represent coordinates within the tetrahedron

Based on this classification, various conclusions can be drawn. A strong allocation of every element to one specific cornerstone of the tetrahedron's base exists for each element, reflecting the original assessment. Here, elements 1 - 3 are assigned to the cornerstone *person*, elements 4 - 6 to the cornerstone *process*, and elements 7 - 10 to the cornerstone *product*. Some elements can only be assigned to one cornerstone, namely elements 1, 2, 8, and 9. The other elements are more diversified and exhibit more dependencies. Especially elements 3 and 6 have a very high dependency among all three cornerstones. All elements have the same value, so there is no ranking between them. Nevertheless, the high degree of interdependence shows that these elements take a special role as binding elements. This is also reflected in the visualization of the means-end relationships in appendix A.5. When taking a closer look at the fourth cornerstone *customer* as the tetrahedron's tip, only element 10 stands out. As mentioned above in the description of the element, the perception of customer integration into the engineering design process has changed significantly over the past years. This strong focus on fulfilling the need of the customer as the maxim is vividly visualized by the fact that it is placed at the top of the tetrahedron. It represents that all activities within the engineering design process shall be directed towards that aim. In line with the description, this representation shall be referred to as the Agility in Mechatronics model, AiM model in short.

5.4.2 Contrasting the perspectives

In the following section, the perspectives chosen in chapter 5.1 are considered and compared to the results of the AiM model. The goal is to explain to what extent the process-related perspective (*Perspective 1: Agile process models in software*) and the domain-related perspective (*Perspective 2: Methodologies in mechatronics development*) differ from the perspective presented by the AiM model (*Perspective 3: Agility in mechatronics development*). This distinct elucidation addresses the issue of misunderstanding from chapter 4.1 in particular.
In the following paragraph, the elements of the process-related perspective are displayed in table 5.3. Afterwards, the elements of the domain-related perspective are displayed in table 5.4. The elements of the AiM model are used to compare the perspectives, utilizing the following indicators for a comparison: Pronounced distinctions are marked by symbol ●○○, slight deviations are described by symbol ●●○, and aspects that are the same or already exist similarly are represented by symbol ●●●. If no statement can be made based on lacking information, symbol ○○○ has been used.

⚠ Remark: Dots represent compliance with AiM model

Analysis from the process-related perspective Table 5.3 represents the process-related perspective in both agile periods, with the elements of the AiM perspective as the reference point. When analyzing the current state, it would be dispensable to consider the *software agility* at this point. However, since some elements of the initial agile approaches are still in use today, they are also listed here for the sake of completeness and comprehensibility. Moreover, by displaying the initial reference point, the modifications to agile scaling frameworks can be depicted (based on the elements listed here).[8]

Software agility period

The software agility period marks the period from about 1991-2006, in which a multitude of agile process models have been developed, as outlined in chapter 2. Their focus was especially on the enablement of small teams that were supposed to become more responsive by close collaboration and communication to quickly create and deploy software. A summary of what these process models aim at accomplishing can be found in the so-called Manifesto for Agile Software Development. For this reason, the Manifesto provides a reliable reference of what was considered an agreed consensus during this period.

[8] However, it should be noted that these modifications are evaluated from the *AiM perspective*, not from the viewpoint of the (original) software domain. For that, other and additional criteria would be used here, but these are not relevant for this investigation.

Table 5.3
Deviations of *software agility* (left) and *scaled agility* perspective (middle) to the elements of the AiM model (right).

Software agility	Scaled agility	Agility in Mechatronics
●●●	●●○	Trust
●●●	●●●	Motivation
●●○	●●○	Learning culture
●●○	●●○	Interdisciplinary collaboration
●●●	●●●	Continuous improvement
●●●	●●●	Short-cyclic gain in insight
●○○	●●●	Systems thinking
○○○	●●○	Connectivity
●●○	●●○	Safety & Security
●●●	●●○	Product excellence

○○○ no statement
●○○ Major deviation
●●○ Minor deviation
●●● Congruence

Human-centered elements

Starting with the human-centered aspects, trust and motivation are mentioned in the fifth principle: *"Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done"* (Beck et al., 2001). These two elements are fundamental components that have been explicitly stated. However, the learning / open error culture is not addressed; rather, short-cyclic action and an increased communication effort and close collaboration are intended to avoid or uncover mistakes. By favoring process-related aspects such as focusing on frequent delivery (DSDM, XP) and embracing change (DSDM, FDD, AM, DDD), these process models have measures to deal with errors. As the open error culture is not explicitly addressed and "learning through collaboration" is only mentioned in DDD, this is assessed as a slight (implicit) deviation.[9]

Process-related elements

When shifting to the process-related aspects, the Manifesto demands *"business people and developers to work together daily throughout the project"* in the fourth principle (Beck et al., 2001). Moreover, (close) collaboration is an important principle in several agile process models (Scrum, DSDM, ASD, DDD). The interdisciplinary part, however, is what is different in this context. The fourth principle calls for an intensified collaboration and communication among people of different background experiences, yet not explicitly among different (engineering) disciplines. This deviation is logical but differentiates team-internal collaboration from interdisciplinary collaboration and is therefore rated as a slight deviation.

The element of continuous improvement is addressed on the human-centered aspect in the twelfth principle by *"reflecting on how to become more effective, and then tuning and adjust one's behavior accordingly at regular intervals"* (Beck et al., 2001). Several methodologies have clearly stated that continuous improvement is a vital principle (Scrum, Crystal, XP, ASD, TDD, DDD). The short-cyclic gain in knowledge is one of the main characteristics of the agile process models in the agile software period. Its incremental nature is stated in the first principle of the Manifesto by *"early and continuous delivery of valuable software"*, whereas the iterative part is mentioned in the third principle by *"delivering working software frequently in short timescales"* (Beck et al., 2001). Moreover, the iterative or incremental aspect (or both) are explicitly stated in almost every process model of that period.

[9] Yet, implicit measures such as, e.g., the two person-rule / four-eyes principle of Pair Programming in XP that relies on trust can be found in the process models of that time. Signs are therefore discernible, but not explicit.

Product-related elements

Shifting to the product-related aspects, the element systems thinking cannot be found, rather its counterpart – the single system, i.e., the actual software product. The process models in the agile software period have their focus on a small number of people, which focus on a single (software) product. They aim for incremental builds and a continuous assembly of individual features piecewise; however, the whole-product focus, as it is stated in the following period, is not yet the focus. No statement can be made concerning the element connectivity, as it is simply not mentioned in this context. This is why it cannot be assessed here. These process models do not specifically address the safety aspect since the focus is on the software product and its functionality, which is ensured by continuous and automated testing (RAD, DSDM). The Crystal methodology is the exception here since criticality (danger to human safety) is one of the decision criteria. Albeit testing being a means of validation to ensure safety, the intent is specifically to deliver something that runs error-free to provide (some kind of) value to the customer. From that perspective, it is considered to deviate slightly. The aspect of product excellence is already present here and can best be described as *high quality by simple design*. Both in the Manifesto and some methodologies, technical excellence and high quality are explicitly required (9th principle, RAD, XP, FDD, AM). At the same time, the aspect of simplicity, which refers to both the process and the product, is increasingly represented (10th principle, Crystal, XP, TDD, AM). Moreover, the strong customer focus is already anchored here in the first principle of the Manifesto.

Scaled agility period

The scaled software agility period marks the period from around 2007 until today, as shown in chapter 2. Here, the focus changed from the application in an originally small team to a greater number of teams (and even the entire organization) working towards a common goal. Besides, it was proclaimed that these process models could also be partially applied outside of the software domain. The challenges associated with that are addressed in chapter 2.3.2. It is interesting to showcase how certain elements from the agile software period have evolved for the following consideration. Primarily, the main focus is on the deviations to the elements of the AiM model.

Human-centered elements

When analyzing the human-centric aspects, one must keep in mind that almost all scaling frameworks are based in part on Scrum or make use of its principles, as described in chapter 2.2.3. Trust is one of the five communicated "key values" in Scrum; however, it receives only limited attention in this context and is not referred to explicitly. Therefore it is rated as a slight deviation. Instead, the element of motivation is pointed out multiple times and considered essential in conjunction with commitment (SAFe, DAD). The element learning / open error culture has not been addressed in the previous period, and it is not mentioned in this period either. However, experimentation and openness (another key value of Scrum) are mentioned in that regard. Learning is the result of experimentation, yet it does not necessarily imply the existence of an open error culture. This is also considered as a slight deviation.

Process-related elements

On the procedural side, the scaling frameworks are very different in terms of scope. Some are very extensive and prescriptive, while others are leaner in this respect. They aim to have several teams interacting with each other and presuppose collaboration to some extent. One aspect that plays a central role is transparency, which is the result of open communication when interacting. Since interdisciplinary collaboration is not strongly emphasized, this element is also rated with a slight deviation. Then again, the element of continuous improvement is explicitly stated in every scaling framework. It is a vital element on the process- and product-related side and aims at both learning and the pursuit of perfection. On the element of a short-cyclic gain in insight, both incremental and iterative aspects are embedded in the scaling process models, which are vital elements. Nevertheless, it should be mentioned that there is a very strong focus on flow optimization, which does not contradict this but shifts the focus slightly. Overall, the short-cyclic procedure and the synchronization of the different teams are important factors.

Product-related elements

When taking a look at the product-specific aspects, the focus has changed to larger (product) scopes, which are handled by a higher amount of people involved. Even though the coordination of many individuals is rather challenging, the *whole product focus* is a key element for all of these scaling frameworks, which is associated with systems thinking. In combination with continuous improvement, systems thinking is a central building block of the process models in this period. This is a decisive difference for the methodologies from the previous period, which strongly focused on a single software product. This strong industry-specific focus is now being discarded. The element connectivity is addressed in the current version of a framework (SAFe). Here it is pointed out as is an important aspect in today's world, and it is mentioned as part of the product/system architecture. Since it is mentioned marginally overall, it is regarded as a slight variation.

As in the previous period, safety is only considered insofar as frequent and automated testing should ensure it. This aspect is present here as well. Security is already receiving partial attention in the newer versions and is considered important (DevOps). From the way connectivity and safety & security are described, the software perspectives to larger product scopes is still present. Such a view is important for developing a mechatronic product and must be combined with the view of mechanical engineering, yet they are not congruent. Therefore, these elements are also rated as slight deviations to the AiM model. The perception of the remaining element product excellence is twofold. For one, a strong customer focus is a central feature in many of the scaling frameworks. As far as the problem-solving of the product itself is concerned, the focus is on a value proposition-oriented result, achieved in a rather lean way. This understanding is more akin to need satisfaction than technical mastery, which is the value proposition in the AiM model. Therefore, it is referred to as a slight deviation as well.

Analysis from the domain-related perspective In this section, the results are compared with perspective 2, classical mechatronics development. This explanation aims at showing how an agile approach differs from a non-agile approach in the context of mechatronics.

Classical mechatronics	Agility in Mechatronics
●○○	Trust
●●●	Motivation
●○○	Learning culture
●●○	Interdisciplinary collaboration
●●●	Continuous improvement
●●○	Short-cyclic gain in insight
●●●	Systems thinking
●●●	Connectivity
●●●	Safety & Security
●●○	Product excellence

Table 5.4
Deviations of the *classical mechatronic* (left) perspective to the elements of the AiM model (right).

●○○ Major deviation
●●○ Minor deviation
●●● Congruence

Trust
●○○

The first aspect of the human-centered view is the element trust, which takes on a central role, as it stands as one of the foundations for a successful application of agile development. This was confirmed by both the experts from the first and second series of interviews. Trust has several aspects: *"Trust within the team (among the team members), as well as trust in the roles within the team (especially when hierarchy is reduced). Furthermore, trust towards and from external stakeholders"* (Interviewee O). Since the term trust is not mentioned in the classical descriptions, "responsibility" is used as a synonym. Interviewee M adds: *"In a classic approach, there is this dichotomy between trust and responsibility. It is often difficult to express trust; control tends to prevail here. The problem here is the willingness to (partially) relinquish responsibility."* The classical understanding of responsibility is the personal liability of the individual for his/her own actions. Interviewee Q mentions that an *"agile approach focuses more on overall responsibility than personal responsibility"* in this context. This aspect is crucial since the responsibility for solving a task or result is thus not measured by the performance of the individual but the team. As a result, the understanding of trust between the two views is still very different so far.

Motivation
●●●

In addition to technical and process-related descriptions, the current edition of the VDI 2221 stresses certain characteristics a developer should possess. Intrinsic motivation is explicitly mentioned among *"experience, wide-ranging interests, non-conformism, openness, a willingness to take risks and flexibility."* However, and that is important to mention, this rather technical description vividly represents the classical understanding of an unambiguous and comprehensive description of requirements. The individual is perceived as part of the development process, and requirements can be placed on that actor. Nevertheless, this description shows that human-centricity is increasingly becoming important in this perspective.

Learning culture
●○○

An open error culture or the ability to learn from mistakes is not mentioned anywhere in this perspective. Risk-taking as a characteristic of a person, as previously mentioned, is required, but what occurs when a mistake is made is not further discussed. In the classical perspective, VDI 2206 states according to Stark and Damerau: *"The reason for the failure of development projects is an inadequately executed requirements development."* Two points stand out in this description. First, failing is perceived here as non-success, i.e., as a loss, not as a trade-off for learning. Second, this description suggests that comprehensive requirements development is the only way to mitigate risks. This contradicts the basic idea of agile development – adapting the solution by continuously obtaining feedback.

Interviewee M mentions that *"an open error culture is often touted but not lived."* As a hint, interviewee C alleges that *"idea validation at the customer's site early in the process without having any prototype is the first step to learn, and thus discard concepts at an early stage, if necessary."* However, this is in strong contrast to the assumption that extensive requirements identification is always the proper approach. Interviewee Q states that the understanding of efficiency must be understood in a different way: *"Since efficiency is the quotient of performance over a certain period of time, the understanding of performance must be redefined here. Performance should also include progress in knowledge and learning."* This understanding has not yet been represented in the classical perspective, which is why it is marked as a great difference.

Interdisciplinary collaboration
●●○

The classic product development literature refers to an interdisciplinary procedure as teamwork (VDI 2221, 2019a). A common understanding is that interdisciplinary product development forms the basis of a design methodology for the development of mechatronic systems (VDI 2206, 2020). However, the term "interdisciplinary collaboration" is mentioned nowhere. This distinction is relevant here because the difference between cooperation and collaboration is the intensity of the exchange. Interviewee H notes that *"there was rarely an exchange between the departments in classic development projects, and aspects were often not discussed because they are simply overlooked."* Interviewee A adds that there is *"often simply a lack of understanding for the other disciplines due to non-existing transparency of the other parties in classic approaches."* Interviewee G mentions that *"ever since agile working methods have been introduced, no more errors have occurred due to a lack of communication."* However, it must also be noted that this increased share of communication is in trade-off for error minimization, which several interviewees confirmed. This distinct difference between simple cooperation of different actors compared to a close collaboration of people who are in a continuous exchange of information is what makes the slight difference here.

Continuous improvement
●●●

The classical view states that *"all activities are usually accompanied by follow-up and learning, which continuously raises the level of information. [...] By continuously reflecting on the problem-solving process passed through, experiences can be evaluated, and improvements can be implemented. This enables a reflection of the problem-solving process and, if necessary, a recording of findings for future processes as well as the derivation of recipes for success (best practices)"* in VDI 2221 (2019a), relating to Albers et al. (2002). The aspect of continuous improvement is a fixed and well-known component, both in agile and classical procedures. The difference here is the frequency, which is, however, particularly due to the next aspect. The continuous improvement of the process is a fundamental component, which is also implemented accordingly in the context of a plan-driven project and anchored in the process models, as in Bender and Gericke (2021), after Dörner (1976). Interviewee L notes that plan-driven development does, of course, align when necessary because *"there is already too much experience in dealing with changes in this respect."*

Short-cyclic gain in insight
●●●

"Iterations are a central feature of product development processes" (Maier and Störrle, 2011; Bender and Gericke, 2021). They are imperative because of the co-evolution of problem and solution, as mentioned by Lindemann. According to Wynn and Eckert, there is even a whole series of different types of iterations, divided into progressive, corrective, and coordinating. *"It is therefore incorrect to speak of a rigid, sequential way of working"* (Bender and Gericke, 2021). Nevertheless, the understanding of short-cyclic gain in insight is different from that of classical product development. This is because *"the most common reason for the failure of development projects is inadequately executed requirements development"* (VDI 2206, 2020). According to the understanding of plan-driven development, a comprehensive requirements development at the beginning of the project is the paramount goal since this ensures safety (VDI 2206, 2020). At the same time, it would be incorrect to say that there are no requirements at the start of an agile project.
The "difference" resides in the conception of a short-cyclic gain in insight. The goal is to produce an increment (pirtype) within one iteration that creates an added value by testing an uncertainty (hypotheses) and thus resulting in a gain in knowledge. According to plan-driven development, *"in product development, epistemic iterations 'should' be allowed"*, and *"iterations in the process 'must allow' for the consideration of new findings"* (Bender and Gericke, 2021).
It is precisely that difference in the formulation that makes the distinction since plan-driven development endorses an incremental approach in the sense of progressive iteration alongside other types of iterations. The understanding of agile development relies on the fact that something presentable (in the form of a pirtype) must be created to achieve a progress in knowledge. Whereas plan-driven development advocates an epistemic approach, the supreme goal of agile development is to force an incremental procedure within one iteration. This difference is rather marginal in theory, but it has far-reaching consequences in practical implementation, which is why this is seen as a slight difference between the perspectives.

As far as the product-specific aspects are concerned, today's state of the art in product development are cyber-physical mechatronic systems, as highlighted in chapter 2.1.3, which was the crucial reason for the actualization of the datasets of the classical perspective. Due to these updates, most of these aspects are in line with those of the AiM model.

Systems thinking
●●●

The concept of system thinking is described in detail in all datasets of the classical perspective. *"The holistic view of the system, the so-called system thinking, represents a core element of the V model"* (VDI 2206, 2020). In the outline of the fundamentals, systems thinking is described (in relation to systems theory) as thinking in functional, structural, and hierarchical contexts or models (VDI 2221, 2019a). Systems thinking is also often referred to as a holistic approach, since *"the value of a solution can only be properly assessed if all boundary conditions, wishes, and expectations are taken into account"*, and this is done *"after a problem and structure analysis by involving all operational areas at an early stage"* (Bender and Gericke, 2021). This is in accordance with the understanding of the AiM model.

Connectivity
●●●
Safety / Security
●●●

The special features in the development of cyber-physical mechatronic systems are the strong networking of highly integrated systems and the ability to change system properties during operation without direct physical access. The interconnectedness of the components of one's own system with each other and the interconnectedness with other systems via a communication layer (such as the IoT), as shown in chapter 2.1.3 as a characteristic of CPMS, are the reason why this topic has come into even greater focus. "*This results in high demands on preventive quality management in the sense of continuous planning and execution of property protection with the aim of establishing system safety and cybersecurity*" (VDI 2206, 2020, p. 10). In addition to these safety aspects added by the technology, the focus of mechatronic systems is already on ensuring reliable and secure handling of the system. Plan-driven development focuses on security through comprehensive and consistent requirements development, which is the basis for evaluating the system for consistency, correctness, and completeness. Both the interconnectedness and the comprehensive view on security are an integral part of the classical perspective.

Product excellence
●●

As far as the concept of product excellence is concerned, the ultimate goal is to find the best possible solution overall from a system point of view. Here, the overall system view is a key driver in particular. However, this aspect is not new, as it has been inherent in engineering for a long time. As far as customer focus is concerned, this is a factor that has been underestimated to date. Interviewee O mentions in this context that *"although we have many excellent engineers who can create technically superior solutions, their weak point is in particular ensuring desirability."* In line with this, *"aligning the product with the customer's desires is not a new aspect, but the strong customer focus is"*, as interviewee N agrees as well. The continual involvement of the customer to ensure his/her satisfaction using pirtypes, which are used so that the customer can already give feedback on specific uncertainties to be tested, is an aspect that has not been considered so far. It aims at tailoring an excellent product to the customer's specific needs, which still represents a slight variation to the plan-driven mechatronics perspective.

Summary of chapter 5

The AiM model consists of ten normative elements, which characterize agility in the context of mechatronics best. The values have been extracted from literature and gathered from empirical data, and have been validated by a large number of practitioners. The overall validity is discussed in the next chapter, followed by a discussion on the results and their implications in chapter 7.

6 Validation

"I've not failed. I've just found 10,000 ways that won't work." - Thomas Edison

In the following chapter, the approach of the work at hand is critically reflected by checking whether it derived valid results. Therefore, the results are checked at three instances: for one, at the beginning of the research approach when outlining the problem statement. Secondly, when the final model is presented. And thirdly, the overall course of action is reviewed as well, which follows the design science approach according to Hevner. In this light, these three validation steps built into the research are presented in figure 6.1.
Chapter 6.1 describes the validation of the initial problem definition between the results from literature (chapter 4.1) and the results from the interview series, as outlined in chapter 5.1.
Next, the model itself is validated in *chapter 6.2*, using the results of the quantitative assessment of the practitioners and the criteria for model development from chapter 4.3.
Finally, the entire procedure in this dissertation is checked using selected criteria in *chapter 6.3*.

6.1 Validation of problem statement

In the first step, the problems in applying agile development in mechatronics were identified based on the findings in the literature and the results of Atzberger et al. (2020a). In addition, the problem areas were confirmed by the interviewees. The results of the challenges are presented in detail in chapter 4.
The relevant group of people (practitioners of agile development in mechatronics in the DACH region) confirm that these problems bear an actual challenge they have not overcome yet; thus, their *currency* is confirmed. This implies that

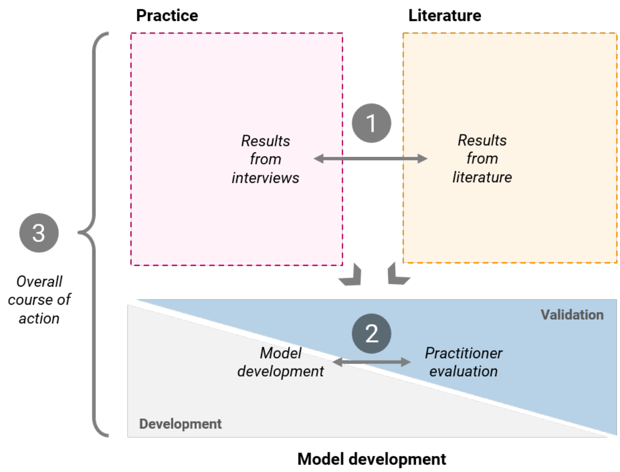


Figure 6.1
Validation indicators built into the research.

the problems identified are indeed challenging and thus of high *significance* for the practitioners as they represent a hurdle that needs to be overcome. Furthermore, (and this point needs to be reflected in both directions) the interviewees have actually experienced these problems, i.e., the persons have a certain experience with the topic and can report from *practical experience* with these issues. In turn, this implies that the interviewees have already gathered a certain amount of experience encountering these issues whilst dealing with the topic and are therefore eligible to act as representatives of the target group.

The general topic addressed here is influencing mindset, as mentioned in chapter 1. In this context, the topic is divided into three sub-areas, which are confirmed by the interviewees as follows:[1]

Understanding

Interviewee A mentions that *"the main driver is a very conservative understanding of agility"* in his opinion. Interviewee I explains this by mentioning that people often ask him, *"So, what's actually new about agile? We've always been working that way!"* This is exactly the issue that Ovesen related to as paradigm perplexity in his work, referring to a misconception of these two different streams of work styles. In the course of validating the individual elements of the model (chapter 6.2.1), one participant even explicitly stated, "Understanding of the agile way of working - very important" as an additional comment in Nicklas et al. (2021).

Interpretation

Regarding the interpretation of agile working methods, interviewee E adds that *"some people don't understand the rationale behind these methods and just apply them bluntly. [...] They use these methods and claim that they develop in an agile way, but they don't understand what's behind it. They also don't understand that a certain goal has to be achieved after a sprint, that it's not about showing some burndown charts, but that it's about delivering something valuable in four weeks [cf. the length of a sprint], some insight, something useful. That's much more important than writing some story maps and throwing burndown charts on the wall."* Interviewee L mentions that *"the Agile Manifesto in its basic features, as it has been formulated, can also be implemented in mechatronics. However, it has to be interpreted in a very specific way based on the given boundary con-*

[1] It should be noted that some interviewees have already been quoted in the problem description in chapter 4.1 to illustrate the extent of the individual problem areas.

ditions." Interviewee H adds that *"the way it is described in the literature, these strict methods, I think it is very rare that these can be applied in that way, i.e., that it is really implemented one-on-one."*

Implementation

As far as the translation difficulties of agility from software to mechatronics are concerned, interviewee L explains that *"these difficulties result from the type of product actually involved. And that's where the possibilities of downstream additions or corrections to development results play a main role. These are the constraints of physicality, the limitations imposed by physicality, i.e., materiality, matter-boundness. [...] Against this background, these material-bound properties give the development team certain limitations in terms of responsiveness, reacting to customer wishes, and reacting to new findings. This way of thinking and the industrial experience of the employees are the cause for these translation difficulties."* Interviewee A replies that, in his opinion, *"the original formulation of the agile Manifesto is a bit too bold, and is now out of date. At that time, it did not come from a service-dominated society but a product-dominated society. [...] In the meantime, however, this service concept has become much more present in the market, and this special focus on value creation has lost some of its usefulness."*

Concluding the problem statement, it turns out that the results from the literature complement the findings from the interviews and are thus in line with each other. Possible solutions for dealing with these challenges from the interviewees' experiences are taken up again in chapter 7.4.

6.2 Validation of model

The findings of this research were evaluated in two complementary ways. For one, the individual elements of the AiM model have been tested and confirmed by a broad range of practitioners in the final step of the embedded design approach. In a second step, the model as a whole has been tested using the validity criteria mentioned in chapter 4. Consequently, the validity of the results of this work is ensured.

6.2.1 Assessment of practitioners

As already mentioned in chapter 5.3, the aim was to validate the model results generated with selected experts in agile development in mechatronics with a broad set of practitioners. The exact details of the questions are presented in table 3.3. The evaluation of the individual ten elements of the AiM model is shown in appendix A.2. The individual elements have been rated according to their importance on a 5-point Likert scale. Since the number of responses per item differs, they are mentioned separately for each item.[2]

3 directions used to evaluate the results

A total of three directions have been evaluated. First, the average (per item) across *all participants*. Second, only people who classified themselves as *experts* (in question D2). And third, people who stated that they had been involved with the topic of agility in mechatronics for *six years or more* (question D1). The respective standard deviation is given displayed for each dataset, respectively. In addition, the entire dataset was checked (using question D3) to ensure that the participants are indeed developing physical products.

[2] Moreover, the option *no answer (N/A)* has been excluded here, which is why the ratings differ.

Overall, every element has been approved preponderantly as it was rated greater than 3 out of 5. The human-centered factors (elements 1 - 3) have been rated as very important (> 4.5) across all datasets. The two elements *connectivity*[3] and *short-cyclic gain in insight*[4] were rated lowest but received the rating "rather important" on average. All other elements were rated >4 and thus received broad agreement across all datasets. There are slight differences in the assessment of individual elements between these three directions, but no clear tendency emerges across all elements.

Connectivity and *short-cyclic gain in insight* rated <4.0

The lower rating of the element *connectivity* could be due to two reasons: For one, it depends strongly on the type of product, since connectivity is inherent for cyber-physical products, but not necessarily for (purely) mechatronic products. Another aspect is the recent update of VDI 2206 to include precisely this aspect of connectivity, which is becoming increasingly important in the development of mechatronic and cyber-physical products, as it has been disregarded for quite some time.

The element *short-cyclic gain in insight* has been rated >4 by experienced participants compared to the "agile experts" and the entirety. This might be because experienced practitioners have already found ways to circumvent the creation of an increment within one iteration, as it is simply not practical due to the constraints of physicality. Instead, interviewee G's opinion is that *"two-week cycles are feasible in mechanical or electronic development. They are not problematic if you abandon the paradigm that you have to have a potentially viable or at least functional system every fortnight. If you discard this paradigm, lead times are not an issue."* In particular, the interviews of the second interview series showed that when discussing the means-end relationships, the short-cyclic procedure is an important part of responding to change, yet it is a very eclectic and comprehensive aspect.

6.2.2 Check against the model criteria

In the last step, the final result, the AiM model, is checked against the criteria defined in chapter 4.3. First, the content-related criteria are assessed, followed by the formal criteria to ensure the validity of the model. As a result, the outcome of this dissertation is examined.

Content-related requirements

The *objective* of this work is to influence the mindset of the individual according to the "mindset approach" outlined in chapter 1. By creating a generic model that considers the different contexts and modes of action of the agile process models, a model has been created that lays the foundation for a target-oriented application of agile modes of action in mechatronics. This objective was achieved, as the detailed explanation in chapter 5.4.2 demonstrates.

The *deficits in practice* are described in detail in chapter 4.1, which subdivide into the three areas of understanding, interpretation, and implementation. The model addresses the topic of understanding particularly, as problems regarding the misinterpretation and implementation are due to a lack in terms of the understanding of agility and its modes of action.

In line with this, the *deficits in theory* in chapter 2.3.3 have shown that there are

[3] Ratings of connectivity across all three directions: 3.76 / 3.89 / 3.79.
[4] Ratings of short-cyclic gain in insight across all three directions: 3.91 / 3.89 / 4.07.

already some process models and approaches characterizing agile development in the context of mechatronics, but an in-depth analysis at the normative level is still lacking. For this reason, a model has been created based on normative elements, which addresses the deficits in theory and practice alike.

Moreover, the *observation scope* was adhered to, as a model was created based on normative elements, the participants involved in the research procedure stemmed from German-speaking countries, and the focus of the research was the agile development of mechatronic products. In summary, it can be concluded that all content-related requirements placed on the model have been met.

Formal requirements

The AiM model also addresses the formal criteria: *Empirical correctness*, which is a measure of correspondence between the model and the actual procedure in the field, is provided because most of the data has been obtained by empirical field research. Moreover, different interviewees have come to the same conclusion, and the results support each other.

In contrast, *formal correctness* describes how precisely the model depicts reality. The high level of approval between the participants in the quantitative study ensures the correctness of the model as the elements of the model are verified by a large number of practitioners. Moreover, as the elements have initially been developed with a small group of experts, who approved them, the quantitative assessment acts as a second iteration; thus, the results have been assured repeatedly. Hence, the results are both formally and empirically correct.

The *purpose-relatedness* of the model is given because the model names ten concrete normative elements that are characteristic for agility in the context of mechatronics, and the model thus frames the topic (research aim 1). These concrete elements can thus be used to differentiate the topic from the other two perspectives of agile development in software and plan-driven development in mechatronics (research aim 2).

The *operability* of the model is granted, as it is simple to interpret and therefore understandable. However, the subsequent application of the model is relatively cumbersome because the generic elements need to be translated. This translation effort is elaborated in chapter 7.2 when explaining the strategies for application in practice.

As far as the *simplicity* of the model is concerned, it is comprehensible due to its ten distinct elements. As the model is of generic nature which has been aligned to the context of mechatronics, which is still relatively broad, a further operationalization of the elements into concrete objectives according to the values hierarchy method needs to be performed. This translation involves a moderate amount of effort, and due to this, it cannot be considered simple in terms of effort. However, this is a trade-off that needs to be accepted, as the model is more generally valid because it is anchored at the normative level.[5]

In summary, the AiM model fulfills all content-related criteria, empirical and formal correctness, and purpose-relatedness. Operability and simplicity compete with the other three, and therefore not all five criteria can be fulfilled simultaneously, as Patzak points out in his definition. Therefore, this model is a compromise, as it is easy to understand in its sole presentation, which was the goal. The

[5] In contrast, a cheap model would already take context-specific (sector, company, other aspects) into account, but then it is not of normative nature anymore, which was the aim of this work.

subsequent transfer to one's own context is nevertheless required for an application in practice; thus, the model is not simple due to the deferred translation effort.

Model characteristics

Furthermore, the AiM model remains a model according to the characteristics once defined by Stachowiak. It shows a representative character as it aims at representing reality. Hence, it is an abbreviated representation of reality, which implies that only certain aspects are emphasized, whereas others have to be omitted. This has been done consciously in the observation scope. The third characteristic is the model's pragmatism, as it attempts to represent agile development on the basis of normative elements for the current time period. This means that the model represents the current state of industrial practice since the values and the interpretation of agility in mechatronics will have changed considerably in the following decade, i.e., another period of time.

6.3 Overall validity

Finally, the entire procedure in the work at hand is examined with respect to validity. Here, seven criteria are considered which are suitable to evaluate artifact development in design science: reproducibility, inter-subjectivity, path dependency, user limitation, internal grounding, theoretical grounding, and empirical grounding (Schmidt, 2019).

Reproducibility

The reproducibility of a design science experiment is guaranteed if the documents and the tested design artifacts are available and thus verifiable (Mettler et al., 2014). The traceability of the development is given since the model conceptualization is presented successively (according to the development steps in chapter 5), and the data for the experimental design artifacts are included in the appendix. However, since the names of the interviewees cannot be published for the sake of data protection, the same people cannot be interviewed, and therefore reproducibility is not possible due to data protection reasons.

Inter-subjectivity

Inter-subjectivity can be ensured when several researchers come to the same conclusion in the same study independently (Schmidt, 2019). Although the author collaborated with several other researchers in the preliminary work (cf. table A.1), this work was carried out by the author alone, as it is common practice for dissertations. Only in the alignment of the agile taxonomy and the values hierarchy method another scholar (the author of the values hierarchy method) was consulted to discuss and verify the alignment of these two artifacts, as elucidated in chapter 5.2. For this reason, inter-subjectivity cannot be assessed here due to the lone working in the course of dissertations.

Path dependency

Path dependency describes the extent to which the model would have a different form or representation if the type of procedure in the model development or the evaluator had been different (Schmidt, 2019). Therefore, this criterion cannot be evaluated here either, but this is characteristic of a qualitative approach in design science, according to Hevner (2007).

User limitation

As far as user limitation is concerned, the basic elements have been determined in a qualitative data collection with experienced practitioners and verified in a second step with additional experts. The fact that the elements of the AiM model are universally valid and not only considered relevant by experts has been ensured by the broad-based quantitative evaluation. This indicates that the individual elements of the model are considered correct for both experienced and inexperienced users. As far as applicability is concerned, the operationalization from generic elements to concrete, context-specific aspects still has to be carried out, which requires a certain amount of experience. This is described in more detail in chapter 7.2. Therefore, the translation effort has to be done by practitioners who are familiar with both agility and their company-specific context. The subsequent application should then be possible for inexperienced users as well. Based on the values hierarchy method, they are able to reconstruct why a certain instrument or tool was selected and which value and objective this is due to if they are unsure. Hence, misunderstandings or misinterpretations are eradicated at the very beginning, which is the objective of the AiM model.

Design theories and artifacts need to be grounded in multiple ways, according to Goldkuhl, who proposes internal, theoretical, and empirical grounding. Therefore, the work analyzes these three types of grounding before concluding on the overall validity of the work at hand.

Internal grounding

Internal grounding means that the design artifact is inherently logical and consistent (Goldkuhl, 2004). The model is structured according to a clear procedure and sets out the bodies of knowledge used. Furthermore, the results from the investigations support the previously identified gaps. The results are concrete answers in the form of means-end relationships (Sub-RQ1), a straightforward model comprising ten elements (Sub-RQ2), which are reflected against the previously used perspectives (Sub-RQ3). The model is coherent because the datasets are mutually supportive, and the interrelationships are coherent, especially due to the use of the means-end relationships.

Theoretical grounding

As far as theoretical grounding is concerned, the design artifact should be consistent with existing bodies of knowledge and embedded in established theories (Goldkuhl, 2004). The model development is based on the alignment of two theories from the literature, which rely on established approaches (DfV on VSD and agile taxonomy on St. Gallen management model). They have been tested and validated in practice for a considerable time by now. Hence, the model is theory-based, and its application uses an approach that has already been tested in practice (values hierarchy method). The model is therefore deeply anchored in theory in any respect.

Empirical grounding

Third, empirical grounding describes the correspondence between intended and actual, observable effects of recommended guidelines for action (Goldkuhl, 2004). The procedure to model development in terms of consulting two rounds of experts and a broad-based assessment of practitioners demonstrates that the problem definition and the solution in the form of the AiM model are empirically sound. Only the practical application of the model and the breakdown according to the values hierarchy method have not been tested using a practical example.

Summary of chapter 6

It can be concluded that the problem definition, the model itself, and the general procedure are valid in various realms. The results show a high degree of trustworthiness and are valid in their entirety; only further empirical grounding is needed, as outlined in chapter 8.

7

Discussion

After presenting the model development and checking the validity of the results, this chapter discusses the AiM model and its implications from different vantage points. *Chapter 7.1* describes the practical application of the AiM model. Since the model's rationale is based on the DfV approach, its operationalization procedure is thoroughly explained.
After that, *chapter 7.2* discusses the usage of the DfV approach itself. As this approach is a fundamental element of this thesis, its application as part of the research procedure is discussed to draw conclusions regarding its usage for further scholars.
The actual solution developed in this thesis, the AiM model, is discussed in *chapter 7.3*. The value of the solution for answering the initially stated research question and its application in practice and academia is assessed.
Finally, suitable preconditions for applying agility in the context of mechatronics are described in *chapter 7.4*. A genuine example is used to highlight which other aspects are necessary in the context of an agile transformation to provide a comprehensive picture in addition to the solution presented here.

7.1 Model application

In this chapter, the basic application of the AiM model in practice is explained in more detail. As the model is based on the DfV approach, the consecutive implementation of the approach and the adaptation to the company's own specific needs is presented here. The procedure is based on the following four steps, as shown in figure 7.1:

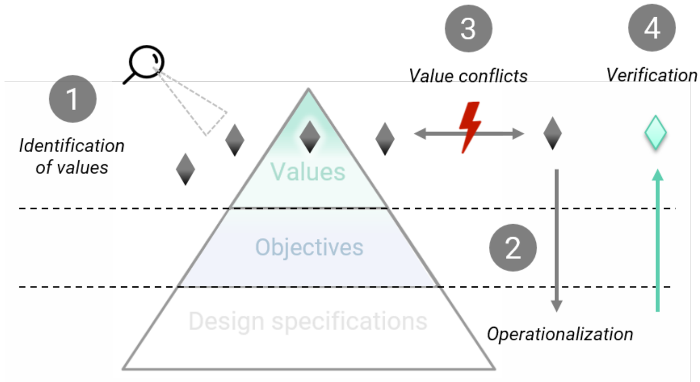


Figure 7.1
Procedure in DfV approach according to van den Hoven et al. (2015).

1 Discovery and identification of values
2 Translation of values into engineering characteristics (operationalization)
3 Unravelling conflicting values and trade-offs
4 Verification of these values

The author of the values hierarchy method, which figure 7.1 is based on as well, states that applying the DfV approach is an iterative process, i.e., a combination of top-down and bottom-up. In addition, it can be considered a method for clarifying the values, norms, and conflicts present in the design process (van den Hoven et al., 2015). The general procedure according to the DfV approach is outlined first, and it is then complemented by an exemplary application of the AiM model in every single step.

1 Identification of values

The discovery of possible values is the first step in applying the DfV approach, which is sometimes quite challenging. Although a list of possible moral values is available from the literature (cf. Frankena (2017)), this can only serve as a starting point for the identification of additional values. A further step is to conduct empirical studies of specific customers or user groups to identify corresponding values. Furthermore, from the company's point of view, it is important to define its own values, as in the case of Volvo, to provide a certain framing. The identification of relevant values for the context of mechatronics has been accomplished by the AiM model, which means the first step for a company is to check the validity or appropriateness of the individual normative elements of the AiM model for a specific technical system (or product). Since the values are valid for the entire context of mechatronics, the relevant values are identified, and, if necessary, additional values such as the company's own values are added while others are removed.

However, it is important to note that there may be one or more additional values / normative elements that might be added to the AiM model even now or in the near future. Furthermore, it is important to keep in mind that due to Stachowiak's pragmatic feature, the ten elements are topical for the current period and may change or even become obsolete in the future.

2 Operationalization

In the second step, the previously determined values are operationalized or specified. Final values are broken down into instrumental values and thus concretized.

№	Design variant	Safety	Freedom
1	Traditional seat belt	Lowest	Highest
2	Seat belt with warning signal	Moderate	Moderate
3	Automatic seat belt	Highest	Lowest

Table 7.1
Overview of process models used for the investigation.

Context-specific objectives are derived from the values, which are, in turn, translated into design requirements, as shown in the *Chicken in the EU* example. In practice, this process step is iterative, as mentioned at the beginning. However, further practical remarks need to be mentioned at this point. While the selection of a value is a first-order value judgment, which represents the selection and fit of the value to the actual purpose (and thus belongs to step 1), *second-order value judgments* are involved when values are operationalized (van den Hoven et al., 2015). They are inevitable when values and objectives are selected since these always have to be adapted to the specific situation. Hence, the context criteria play a decisive role when applying the values hierarchy method in practice due to these underlying second-order value judgments. For more information on second-order value judgments and their implications, see (van den Hoven et al., 2015, p. 168ff.).

In relation to the application of the AiM model, the values identified and determined to be necessary from step 1 are used here. As in the *Chicken in the EU* example, objectives are now defined based on the values, and these are in turn specified in the design requirements. In practice, this process of operationalization (top-down) and generalization (bottom-up) takes place in an interplay until an optimum is found.

3 Value conflicts

Although value conflicts can occur already in step 1, it is common while interacting iteratively that further conflicting value interdependencies become visible, which is why this step is carried out third, according to the guidelines. There is a complete chapter on this in van den Hoven et al. (2015), which is why value conflicts in engineering design are dealt with in particular. If a value conflict occurs, i.e., two or more values are opposed (e.g., as in the iron triangle), a trade-off must be achieved to resolve the conflict. For this purpose, the values need to be *harmonized*, i.e., aligned as closely as possible (cf. balancing of values). This can be achieved by the use of satisficing to tackle the value conflict. *Satisficing* stands for a combination of satisfy & sufficient and tries to find a solution on a rather quantifiable level (van den Hoven et al., 2015).

To illustrate this procedure, a vivid example from the automotive industry is used – the seat belt, which Volvo developed first, as safety is considered one of the company's values. Here, three design variants are to be evaluated between the values of safety and freedom, as displayed in table 7.1. It should be noted that this is a much-abbreviated representation. The original example can be found in (van den Hoven et al., 2015, p. 92f.). Although this bottom-up comparison contrasts the design variants with the values, this simplified example illustrates the issue of value conflicts. By using further criteria, these variants can be made quantifiable and comparable to solve this value conflict as a tradeoff. However, this again involves second-order value judgments.

When shifting to the AiM model, one value of the model could cause a value conflict with another value, or even conflicts among values of the AiM model might occur, such as *error culture* versus *safety & security*. Solutions ensuring the efficacy of both (e.g., in production) from Volvo or Toyota (cf. chapter 1 and 2) show that this value conflict can be solved, which is why the safety belt was used as a direct application.

4 Verification of values

The final step is to verify the implementation of the values as a means of achieving the intended goal. The aim is to check to what extent the design requirements embody the values and align them with the overall result. This step is also part of the iterative process between synthesis and analysis in the problem-solving process. It has to be carried out particularly in the company's own context and becomes important when new or changed values are considered.

Although an application of the AiM model has not been possible due to the lack of context, the development of the model followed the same pattern. The verification of the individual values of the AiM model was carried out by the participants of the second interview series.

7.2 Assessing the DfV approach

In this chapter, the Design for Values approach is contrasted in particular. First, the use of DfV in the context of this dissertation is considered in more detail, and other possible approaches are presented. Next, the advantages and disadvantages of the DfV approach are presented before concluding with a review of the use of DfV in general.

Two aspects are novel to this work concerning the DfV approach. For one, to the best of the author's knowledge, the DfV approach has not been used as a research method before. In this work, the DfV approach, combined with the Taxonomy for agile development, forms the central basis of the research method, which has been realized within the bounds of an embedded design approach. Thus, while following a mixed-methods approach, the underlying object of research was analyzed and characterized using the DfV approach. Hence, its application forms the groundwork for potential future use as part of a research approach.

Secondly, it should be noted that the DfV approach moved beyond its actual scope. This work forms the foundation for the first application of DfV outside the actual scope of determining product characteristics. By analyzing the process characteristics of agility in mechatronics, the product of interest was considered any mechatronic product, not a specific one. Thus, the procedural modes of action have been studied, and all relevant elements have been included in the AiM model. Due to this, DfV can also be applied in additional areas, as this work has shown.

Alternative procedures

In addition to using DfV in an embedded design approach, the research question could have been carried out using action research or by applying the lead user method. The application of the action research method, which can be well combined with the design science research approach, requires that the researcher is part of the research process and thus actively influences the procedure. Due to the lack of access to necessary resources, that was not practicable in this constellation, which is why this path has not been selected deliberately.

Another approach would be to use the lead user approach as a research method. Schmidt has shown in his work that the lead user approach produces very convincing and reliable results in a highly hyped topic, as only actual experts are consulted. However, since the lead users had been identified by Schmidt for a very similar and adjacent topic area (cf. constraints of physicality in agile product development) only two years earlier, the same work was not repeated in a slightly modified form. Instead, the existing contacts were used, and since these people are all still actively involved in the topic (due to their active participation in the expert network APPe), the experts for the second interview series in this work were selected from this circle of lead users, thus creating a synergy of both approaches.

In the implementation and application of the DfV approach, some aspects went quite well, yet others were rather challenging. Thus, to give a brief overview, the advantages and disadvantages that need to be considered are summarized here.

The following aspects are considered *beneficial*:

Advantages

Neutrality ✔

By using the DfV approach, a particular issue is illuminated on a normative level. This is characterized by the independence of context and the genericity of values. In this way, a factual situation is described neutrally and only in the following steps subjectivized due to second-order value judgments. In addition to comprehensibility, this also creates credibility, as subjectivity is eliminated on the normative level.

Strategic positioning ✔

The DfV approach can be used to derive and define concrete measures from one's own values. In this sense, it is a strategic approach that links business development and product development. Furthermore, in combination with the St. Gallen management model, it can be a sound way of addressing potentials in the early phase of product development.

Transparency ✔

The result of the application of DfV primarily provides an explanation of the cause-effect relationships. This transparency, which results from the illustration of the interrelations, then ultimately results in clarity. Since communication based on corporate values has increased considerably in recent years, the DfV approach provides concrete recommendations for action and translation, which counteracts ambiguity in decision-making and ensures comprehensibility.

Mindset support ✔

A central aspect, especially for this work, is the support of the mindset. Due to the objectivity of the approach as well as its transparency, the modes of action can be communicated and conveyed in a comprehensible way. This is especially important for such a hyped topic as agile development, as it counteracts the guru problem.

Disadvantages

The following aspects are rather *challenging*:

× **Interpretation of terms**
In particular, the understanding of what characterizes a value is diverse in practice. For this reason, in the course of the first series of interviews, it was often necessary to use the term "objective" to obtain answers and then return to the actual values in the further course of the interview. Another aspect of consideration is the different meaning of certain terms in different cultural contexts and their (mis-)translation.

× **Abstractness**
Another aspect derived from the first term is the ability to abstract, which is necessary to discuss values on a normative level. This was a challenge, especially for participants in the first series of interviews. In this context, it is not only the ability to abstract from the actual topic but also from one's own company context and experience background that is meant. Since this very theoretical elaboration is very distant from practice, concrete examples and applications are helpful for practitioners.

× **Context specification**
Another aspect that is important when using the DfV approach is the need for a certain contextual setting. It is essential to define this boundary in advance and adhere to it in the further course in practical applications. This was achieved by aligning the taxonomy to the field of agility and made more concrete by narrowing it down to mechatronics, as shown in figure 2.17. However, this is much easier to implement in theory than in practice, which is why this is explicitly mentioned as a challenge.

× ***Expert knowledge needed***
The last aspect is particularly important for the development of a generic model such as the AiM model and thus does not apply in general.

The means-end relationships were used as a vehicle to identify the correlations and links between the different elements and discuss the effects of agile development at the normative level. However, practitioners require a high level of expertise and the ability to abstract in order to be able to relate the individual aspects to one another and to contribute to the topic actively. For this reason, practitioners with the most in-depth knowledge, the lead users, were consulted for the second interview series.

Summary on DfV approach

The result of the DfV approach is a clearly defined derivation of product characteristics based on predefined values. The approach creates transparency within the development team and beyond, as the links between the defined values and the derived measures are visible. If a certain topic faces a high degree of hype, a consensus is needed based on criteria that are as objective and clear as possible. Hence, when the challenges outlined in chapter 4.1 are taken into account, applying the DfV approach as a research method can be recommended. The use of the approach beyond the mere definition of product properties can also be successful, as shown in this work. Overall, Design for Values is a powerful approach that suits the needs of the current times.

7.3 Value of solution

Every increment aims to create a certain value. Hence, with the AiM model as a design artifact being the final increment of this work, its value can be assessed in the following three realms. First, the model's value is described based on its assessment in accordance with the research questions. Second, the value of the model for industry is elucidated before highlighting the value for academia in the third step. Third, the suitable preconditions for agility in mechatronics are summarized and enriched by best practices from industry.

7.3.1 Value contribution in accordance with research aim

Starting from the initial clarification of the research objective, the following section reflects the value contribution of this work measured against the research questions and sub-research questions set up initially. In line with this, the first part is concluded by discussing how the value-based model addresses the initially stated claim of the work at hand.
The overall aim of this thesis has been stated in the research question: *"How can the mechanisms of action of agility be harnessed in the context of mechatronics?"* In order to answer this research question, it has been divided into three sub-research questions. Firstly, the mechanisms of action for agile development in mechatronics are presented; secondly, the essential values that describe agility in the context of mechatronics are identified; and thirdly, the result is reflected in relation to other perspectives.

Sub-RQ 1

The first sub-research question, *"What are the mechanisms of agile development in mechatronics?"* therefore deals specifically with analyzing the modes of action. Since *"a holistic explanation of why agile development works does not [yet] exist"* (Schmidt, 2019, p. 39), this work examines the complex and interwoven relationships using the first artifact, being the *means-end relationships*. This artifact offers the following advantages: The findings are presented in the form of means-end relationships, i.e., one can clearly reconstruct which aspects are influenced by others and what results follow based on these. This logical presentation thus promotes understanding, as it functions as a "white box" representation showing all interrelations and not only the known inputs and outputs without explaining the mechanisms of action. In addition, these compounds are based on practitioners' experience and have been validated by practitioners; this implies that their own target group understands their validity and comprehensibility. This is particularly important for inexperienced users, as they can easily grasp and interpret the connections. Moreover, this step-by-step sequence provides clarity and transparency, which counteracts the aforementioned guru problem by showing exactly which steps are interdependent.

Sub-RQ 2

The actual design artifact generated in the course of this dissertation is intended to answer the second sub-research question: *"Which values are characteristic for agility in mechatronics?"* Ten elements have been identified that characterize the AiM model on a normative level based on values. Their validity and completeness

have been ensured in the last step of the embedded design approach. Although all elements were rated as important, two elements fell below the distinct approval threshold of >4.0+ out of 5 (on a 5-point Likert scale rating, as displayed in table A.2). These are the two elements *connectivity* and *short-cyclic gain in insight*. Although a brief explanation of the possible reasons has already been provided in chapter 6.2, some interviewees also addressed one essential aspect, which is the central question of what exactly is meant by a mechatronic product in 2020. Through the connection to the Internet of Things (IoT), the previously autonomously acting machines will become cyber-physical systems, according to figure 2.7. In the context of this dissertation, the term mechatronics is therefore used as an umbrella term for all further developments of mechatronic systems towards CPS, as explicitly mentioned in chapter 2.1.3. From this point of view and the specific reference in VDI 2206 (2020) that increasing connectivity was the main reason for updating the guideline, this element is of central importance for the model. In particular, the experts in the second series of interviews also mentioned this aspect as indicative of the coming decade, since as interviewee B mentions, *"we as a company are evolving from a machine supplier to a system supplier."*
The second element short-cyclic gain in insight is often misinterpreted, especially in the context of mechatronics. Particularly inexperienced users of agile methods are very much oriented towards the Manifesto and focus on delivering "physical" prototypes within a sprint very strictly. This is, however, not manageable due to the constraints of physicality, as Ovesen and Schmidt point out. However, creating physical prototypes is crucial, especially for engineers, because as interviewee A noted, *"what I can't hold in my hands, doesn't count."* Interviewee C adds that this *"condition of being able to experience something new"* still plays a central role in this regard. In this context, interviewee G states that *"this is also a learning process and that short-cyclic progress in mechatronics is feasible if you abandon the paradigm of generating a physical prototype every 2-4 weeks, but something value-creating in the form of an insight instead."* The means-end relationships show that this short-cycle timing tends to take a back seat, as it includes several aspects and is not obvious if not explicitly stated, but according to interviewee Q, *"this iterative timing is extremely important."*
Thus, it can be concluded that there are understandable reasons why these two elements do not receive full agreement (<4/5) from all participants; they are, however, nevertheless important and thus central elements of the AiM model.

Sub-RQ 3

The third sub-research question *"How does agile development in mechatronics differ from agile software development? How does it differ from classical, plan-driven development?"* relates the results of the AiM model to the other perspectives. The aim is to provide a clear and comprehensible explanation of how and to which extent certain elements differ in the various perspectives or what they have in common. This differentiation is particularly important as it shows how agile development in the context of mechatronics differs from its application in software. Furthermore, it is shown which elements in classical (plan-driven) development (which has developed further over the years and has come closer to the spirit of the ages as well) differ from those in the AiM model. Based on the interviewees' experiences, this presentation is intended to offer as neutral a representation as possible of which differentiating features still exist and thus serve as a basis for explaining which aspects are new compared to the past and what exactly constitutes an agile compared to a classical approach. Furthermore, the entire work shows which elements the current process models use and presents in a value-neutral way which elements are also more or less relevant in the other perspectives. It also indicates which elements or aspects have not yet been sufficiently considered or understood differently.

Answering the claim

The claim of this work is to influence the mindset of the individual as initially introduced as "mindset approach" in chapter 1 (Gericke et al., 2021). In order to achieve this, the mechanisms of agility have been transferred from the context of software to mechatronics employing a value-based approach. The use of values at the normative level was chosen to achieve a sustainable and comprehensible context transfer. As a result, the individual problem areas, as shown in chapter 4.1, did not have to be addressed individually but could be directly tackled at the root cause. The AiM model, with its ten elements, provides a value-neutral definition of what characterizes agility in the context of mechatronics due to its interrelationships and characteristic elements. Finally, it is still necessary to provide assistance in the operationalization of values from the normative to the operational level, which is elaborated in chapter 7.2.

7.3.2 Value for industry

In order to assess the value for industry, the *relevance for industry* from chapter 1 and the *deficits in practice* in chapter 4.1 are consulted.

Relevance for industry

As far as the relevance for industry is concerned, studies show that current agile methodologies and scaling frameworks lack sufficient context-specific alignment due to high degrees of adaptation (Atzberger et al., 2020a). For this reason, the objective of the AiM model was to address this context shift explicitly. To overcome this problem, several companies (2020: 39 %), according to Atzberger et al. (2020a), are already creating their own process models. This significant share reflects the high need of companies for a context-specific solution. Schmidt proposed that an agile method specifically tailored to the needs of the hardware context, which provides practices from well-known agile methods (such as XP or Scrum), would have to be extended by other hardware-specific practices. These additional hardware-specific practices could then be seen as a supplement, like a "Scrum for Hardware" add-on. Although this is correct, it should be noted that Schmidt's work considered the constraints of physicality in particular and not the topic of mindset. Although this addition is appropriate for the technical-physical implementation, there is still no adequate context transfer, and the deficits in practice are not adequately addressed.

Deficits in practice

The term "mindset" is used to cover the areas of understanding, interpretation, and implementation, which are summarized as deficits in practice in chapter 4.1. A clear understanding of the objective of all stakeholders is needed to address the difficulties in understanding. This, in turn, entails creating a certain awareness and appreciation of the concerns and needs of the other domain. Interviewee A describes this as follows: *"The understanding of software development as a production process is more distinct in a hardware-dominated development department. Conversely,* *in a software-dominated development department, the understanding of long-term (production) times in hardware development is non-existent. This awareness for the other domains has to be established in the first place."* In this regard, the element interdisciplinary collaboration of the AiM model comes into play as close collaboration among domains creates this understanding among different disciplines.

To address the difficulties in interpretation, a certain degree of unambiguity is crucial. This unambiguity does not only stem from clarity in their description but also from accuracy of fit. In this regard, interviewee A mentions the following: *"In my opinion, the original formulation of the Manifesto falls a little short of the mark and is out of date nowadays. At that time, it did not come from a service-dominated but a product-dominated society. Large companies pushed their products on the market, and there was nothing else. So they could also place whatever they wanted on the market, and people would still buy it. In the meantime, however, this idea of service has become much more present in the market, and this special focus on value creation has lost some of its usefulness."* Thus, the lack of further development is a central factor that fuels the difficulties of understanding, especially for a different context. If the mechanisms of action are understood and clear, the demand for a physical prototype at the end of a two-week sprint is reinterpreted and adapted accordingly in the context of mechatronics, as previously explained by interviewee G.

Thirdly, the application challenges can be avoided if the process fits and is tailored to the context. Although tailoring processes is a common measure, process adjustments must also be continuously monitored, and, in addition, the suitability of the process must be checked as a first step. This fitness check was carried out in the course of this work on a general level. Interviewee R (because of being a lead user) mentions that they have already done something similar in their company: *"The context transfer that you did in your work is quite similar to what we have been working on in our company for the past year [2019]. We have chosen specific days, staffed a truly interdisciplinary team with people from HR, product management, engineering, process management, and agile methodologists, and went out of the company into a 'think tank', specifically for this. Our goal was to find out why the measures and actions, which make total sense on paper, didn't work in practice."* This shows that, on the one hand, the need among companies is quite high and, on the other hand, adaptation is a real problem due to the current non-alignment of agile methodologies when lead users spend so much workforce on resolving this issue.

7.3.3 Value for academia

In order to assess the value for academia, the *relevance for academia* from chapter 1 and the *deficits in theory* in chapter 2.3.3 are consulted.

Relevance for academia

As outlined before, a proper definition is needed that explains how agility is to be understood in the context of mechatronics. Moreover, this definition has to be scientifically sound yet also easy to comprehend to counteract the guru problem. The approach taken in this work addresses both aspects, as it is a combination of two scientific models which have been merged. Moreover, they are simple in their structure and straightforward to understand. Since there are currently a mere amount of approaches available that pursue an agile procedure in mechatronics, the AiM Model represents a foundational model that further procedure models can be built upon and derived in a context-specific manner. Although it is not a process model itself, it provides a theoretical foundation for the essential elements that an agile methodology for mechatronics should take into account. Furthermore, uncovering the mechanisms of action by using means-end relationships is a key factor that adds value to practitioners' understanding and the knowledge base, as it provides epistemological value.

Deficits in theory

There is only a very small amount of prior work that examines the mechanisms of action of agile development at the normative level, as shown in chapter 2.3.3. Due to the use of the DfV approach, the identification of values is, therefore, a scientifically substantiated approach that has already been tested in practice. However, it must be mentioned that this approach (in its original version) is intended to identify values for a product to be developed, not a process. This adaptation is nevertheless permissible, as the basic logic of the procedure is not changed by it, which was also confirmed by a co-author of the DfV approach. Furthermore, the alignment of the values hierarchy method and the Taxonomy for agile development was discussed with the author and found to be valid as well. Therefore, the use of the DfV approach for identifying values in its modified form for process characteristics will be incorporated into the knowledge base.

Another deficit is the lack of a representative model of agility for the context of mechatronics, or an agile method for mechatronics, as Schmidt referred to it. According to current studies (Schmidt et al., 2019; Atzberger et al., 2020a), the process models and frameworks that have been used in the context of mechatronics so far all rely on methodologies that were originally developed in the context of software. Although there have been minor revisions in the methodologies, there has not yet been any contextual adaptation, whether at the level of the fundamental philosophy in the Manifesto or the process models derived from it. Thus, the AiM model represents a model exclusively designed for the context of mechatronics for the first time. Contrary to other models, the model thus fits the context because it has been specifically designed for and validated within it, whereas the other methodologies originating from software have to make concessions.

A third deficit is the so-called "paradigm perplexity" according to Ovesen, which describes the clash of different ways of thinking within a system and the associated uncertainties. It describes the flow of elements from agile development into the previously practiced, classical development and the resulting confusion among some of the people involved. As it is based on the DfV approach, the development of the model and the operationalization of values are of great interest for both practice and theory. Overall, the comprehensibility of the measures based on operationalization is of high value, as this clarity directly influences the mindset positively. This will be explained in more detail in the following chapter.

Going back to table 1.1, the scope of the work is outlined. From a methodological point of view, a model has been developed that is highly abstract and thus generic in nature, as it is located at the normative level. This offers the advantage that it can be applied across all sectors in the mechatronic domain. Nevertheless, it is specifically geared to the context of mechatronic products as it has been specifically designed for that, compared to the already existing methodologies and frameworks. It contains ten elements that are straightforward to under-

stand, and whose mechanisms of action are comprehensible due to the means-end relationships.
Interviewee R concluded the interview by adding the following: *"The individual terms we have just discussed are actually not a novelty, otherwise we could not have a discussion about them, but you have re-arranged them and given the topic a whole new framing. That's what you scholars do in your dissertations, and that is the actual added value."*

7.4 Suitable preconditions for agility in mechatronics

The *sweet spot* for applying agile development has already been briefly mentioned in chapter 2.4, and the topic areas in the context of agility for mechatronic products are also presented in figure 2.13. For an appropriate application of agility in mechatronics, further aspects are essential, whereby the adaptation of agile values to one's own corporate values represents an important building block. In order to sustainably establish agile development within the company, it is important to pursue an agile transition after the first successful pilot projects. This denotes the implementation of agile working methods in a specific department. For agility to work in the context of mechatronics, the following three steps are important, which need to be considered in addition to the adaptation of values:

1. Alignment of process organization
2. Adaptation of company structure
3. Establishing an appropriate corporate culture

These particular topics are briefly addressed here, and each is supplemented by a genuine example from practice for the sake of illustration. While this demonstrates a possible way of adapting these individual steps, it should only be understood as a recommendation for action that always needs to be adapted to the company's own context.

1 Alignment of process organization

The focus of this work is on identifying the relevant values for the engineering design process. The AiM model provides initial normative elements that must be adapted to the company's own context (see chapter 7.2). Based on the company-specific adaptation of the values, a transfer to the process models must occur, i.e., the engineering design process has to be adapted accordingly. In the state of the art, the different process models for both classical (chapter 2.1.2) and agile product development (chapter 2.2.3) have been presented at the beginning, supplemented by those used in the context of agile development of mechatronic products.

Classification of agile process models in PDP

As noted in the introduction, the first approach was to transfer agile methodologies from software to mechatronic product development. In this context, Böhmer et al. has depicted the suitability of agile methodologies along the product development process, starting in the fuzzy front end, as shown in figure 7.2. This classification of the agile process models can be used as a reference point that compares the potentials of the individual methodologies in a rough overview to

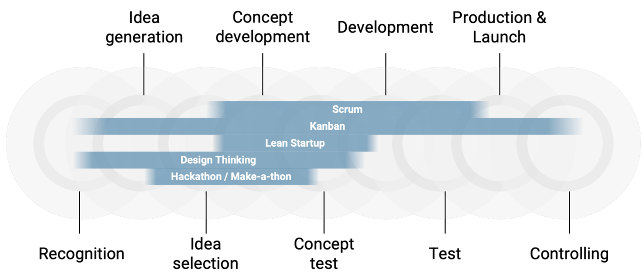


Figure 7.2
Agile process models along the product development process according to Böhmer et al. (2015).

draw one's own conclusions for adaptation to the company context.

Although a mere transfer of the methodologies into the product development process is possible, the challenges in mechatronics are not dealt with here. As addressed above, these are known as constraints of physicality (CoP), which describe the differences that must be taken into account in developing physical products compared to software development (cf. figure 2.12). Furthermore, one must recall that a wide range of technical products (and systems) in the sphere of mechatronics exists, whereby the more complex the product, the greater the impact of the CoP due to the magnitude of the interlinkages (Schmidt, 2019). In addition to the system complexity, the technical-physical dependencies have to be taken into account in the development of mechatronic products. Here, certain work packages cannot be moved or assigned according to changing priorities since a physical product cannot be "cut" arbitrarily, i.e., certain steps in the development and prototyping simply have to take place one after the other due to the laws of physics, as interviewees E and G point out. According to interviewee H, *"in mechatronics, at some point there is this hard border, in our case the M2 [milestone 2; transfer point to series development], when the parts for the prototypes have to be manu-factured. [...] Therefore, a certain pre-planned procedure is necessary in the context of mechatronics for products with high system complexity due to their high number of interfaces."* In his case, a prototype *"easily costs a million [Euros], and its manufacture takes several months."*

Nevertheless, the use of agile process models is possible, albeit with adaptations. This combination or major adaptation of methodologies to one's own context is referred to as (a company's) own process model, such as, e.g., the Spotify model. Interviewee G states that *"the agile development process in his company follows a Scrum-based procedure, with the additional position of team leader and a classic project plan in the form of a Gantt chart."* If one follows the Scrum textbook, any variation of the procedure "is then just not the same as Scrum," according to the Schwaber and Sutherland. However, since an adaptation to the context criteria is important, it does not have to be "Scrum by the book" for an application in the context of mechatronics. Interviewee G describes this using the example of what constitutes the actual purpose of an increment, which *"must not be seen as a physical occurrence, but as a short-cyclic gain in knowledge."*

Schmidt thereby outlines the implementation of an iterative procedure in the context of mechatronics with the use of two different types of iteration, as it can be seen in figure 7.3. While the exploration iteration follows the original prin-

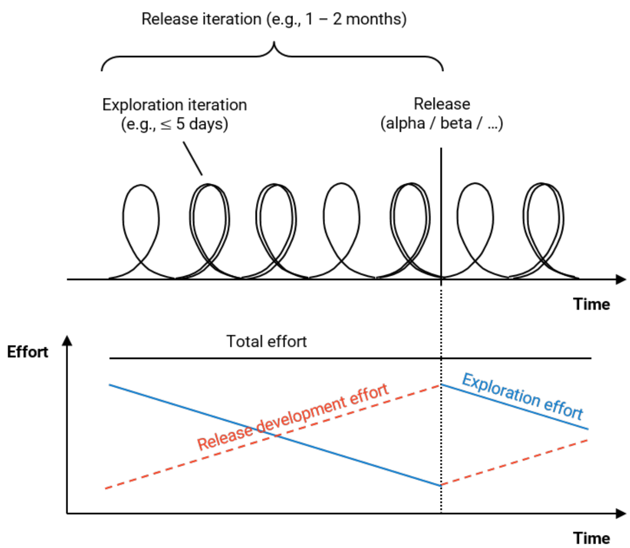


Figure 7.3
Iteration types for the context of physical product development according to Schmidt (2019).

ciple from the Manifesto and ensures the short-cyclic generation of knowledge, release iterations represent the cycles for the creation of a physical prototype, which lasts several exploration iterations due to the CoP. If one follows Schmidt's suggestion that a *pirtype* is sufficient as an increment of an exploration iteration, then agility can be implemented in mechatronics with ease (Schmidt, 2019).

Process adaptation in practice

A common solution approach from practice is the combination of a phase-based procedure (such as Stage-Gate) with individual selected agile procedure models in different phases, as it is envisaged in ASD or Agile Stage-Gate. A practical implementation of this is shown in figure 7.4. Here the innovation process is shown in different phases through the use of various agile process models. The phases are based on the product development process, but the focus is geared towards the three areas of desirability, viability, and feasibility in the early phase of the development process. Interviewee O explains that *"we as a company have excellent engineers who also develop great products, but we have distinct weaknesses in understanding the customer benefits and the profitability of our solutions at an early stage."* In this context, he refers to the Design Squiggle[1] as an illustrative representation of the fuzzy front end of a design process that could be used analogously to visualize the diminishing of uncertainty. To counteract this, the agile development process is oriented precisely to these areas. Design Thinking is primarily used to understand the customer's problem and identify possible customer wishes in the desirability phase. Next, viability is tested using Lean Start-up to validate initial findings for possible solutions from an economic perspective. As technical feasibility is one of the company's strengths, this is considered last using a Scrum approach to generate the first tangible prototypes.

[1] „The Design Squiggle is a simple illustration of the design process. The journey of researching, uncovering insights, generating creative concepts, iteration of prototypes and eventually concluding in one single designed solution. It is intended to convey the feeling of the journey. Beginning on the left with mess and uncertainty and ending on the right in a single point of focus: the design" (Newman, 2002).

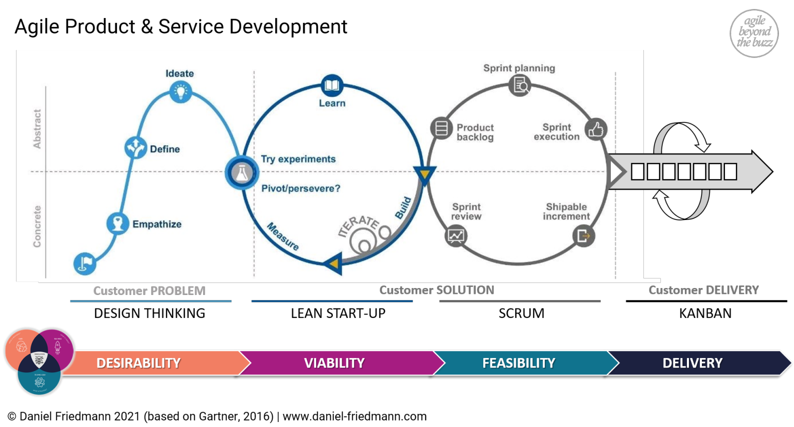


Figure 7.4
Exemplified use of agile process model. Image licensed under Common Creative and used with permission of the author.

These three areas are covered in the first phase of the product development process before the next step is to switch to Kanban when it comes to delivering the solution to the customer. This is illustrated once again at the end of this chapter. As initially outlined, the individual phases are characterized by a Stage-Gate approach, *"as it is simply indispensable in mechatronics"* according to Interviewee O. In this approach, however, the quality gates are called innovation maturity gates to focus more on the maturity of the innovation. Interviewee C mentions that ascertaining technological maturity is important in developing mechatronic products, which is why the concept of TRL (technology readiness levels) is used in their development process.

In conclusion, it can be asserted that the design of the development processes strongly depends on the contextual framework conditions. In this regard, Böhmer distinguishes between active vs. passive agility. Here, passive agility describes that *"the development teams know 'what' to do, but not 'how' to implement it"* (Böhmer, 2018, p. 130). In contrast to this, *"innovations are characterized by high uncertainty about the problem and solution space that must be explored actively, agile teams perform pilot studies to understand 'what' the user needs while implementing potential solutions"*, which characterizes active agility (Böhmer, 2018, p. 130). These different characteristics correlate significantly with the context, as both active and passive agility have their sweet spots, as (Böhmer, 2018, p. 130) and (Schmidt, 2019, p. 140) show.

(2) Adaptation of company structure

Another important topic is the change or adaptation of the organizational structure. Although no recommendations for action are provided, the practical example is intended to show how the structures have been successfully changed so far. A central aspect of agile development is establishing cross-functional teams, i.e., the composition of a group of people from different departments and professional backgrounds. This prerequisite by itself poses a great challenge to many companies in the field of mechatronics, as a team is often comprised of people from the same technical backgrounds.

An illustrative change in the organizational structure can be found in, e.g., the *Spotify model*, which was developed in the context of software and is displayed

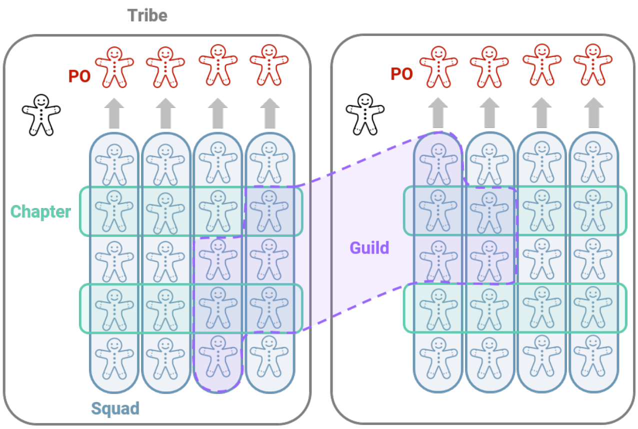


Figure 7.5
Team composition of the Spotify model according to Kniberg and Ivarsson (2012).

in figure 7.5. It focuses on the interaction and coordination of several teams (a tribe comprises several teams), whereby it is strongly oriented towards a Scrum team at the team level (corresponds to a squad, blue circle). In addition, there are so-called chapters for inter-team communication, in which certain experts are grouped. A guild represents a "community of interest" that is dedicated to a specific topic, even across several tribes. An example of this is the Agile Coach guild. Although this resembles a matrix organization, the basic understanding is strongly focused on delivering something of value. A detailed description of the whole model and its components can be found in Kniberg and Ivarsson (2012). The example is intended to demonstrate that even in the context of software, pure adoption of agile methodologies does not always lead to the desired results. The Spotify model with its modifications vividly depicts that adaptation to one's own context and products are inevitable.

A further aspect is the establishment of flat hierarchies. This point implies a complete restructuring of the organizational structure, as existing structures have to be changed and intermediate levels consolidated, which also entails the transfer of responsibility. The example from practice illustrates this once again.

Structural adaptation in practice

"First, the introduction of agility was started in a pilot project, and in the next step, additional teams were transformed after initial context-specific adjustments" (Interviewee O). However, according to him, the following effect occurs after a certain number of teams, which he compares to the following metaphor: a Formula 1 car stuck in a traffic jam. If one's own team or the neighboring areas are changed to agile working methods, this works well as long as they are in the same cosmos. However, when working with classic, non-transformed teams, exactly those issues occur that correspond to the metaphor of the Formula 1 car. In order to solve this "traffic jam", a change in the organizational structure is necessary. In this example, six different business units with functional silos have been divided into more than 50 individual entrepreneurial units with cross-functional teams.

As shown in figure 7.6 on the left, the organizational chart of the original structure is divided into five levels. This organizational structure was restructured based on the *flight levels model*, which offers the advantage that several products

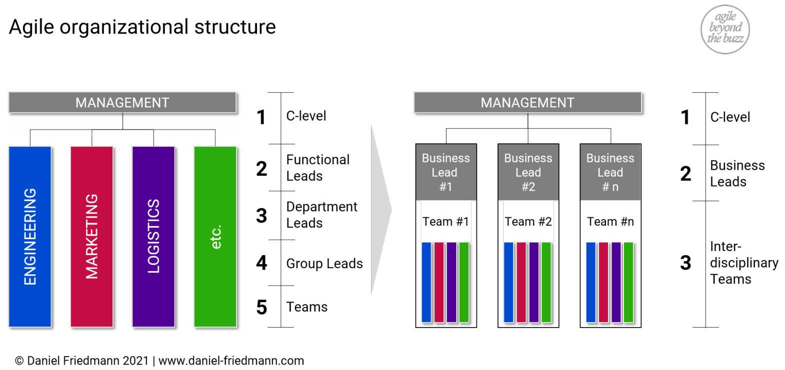


Figure 7.6
Change of organizational structure. Image licensed under Common Creative and used with permission of the author.

can be processed that only exhibit few inter-dependencies. The original five levels with the corresponding functional units are converted into a simplified model based on three levels. The functional silos are divided into two different types, namely *purpose teams* and *expertise teams*. The focus of value creation lies with the purpose teams, whose task is to push a specific product group. The expertise teams bundle competencies and can be called in as needed according to specific expertise. Thus, the core competencies are not all bundled in one place but are anchored in the individual teams in the corresponding purpose teams, which can act "more agile" due to their small size.

Nevertheless, a trade-off has also been made in the course of this restructuring. Interviewee O explains: *"Our goal in changing the organizational structure of the business unit was to focus on customer-centricity, i.e., to sharpen our focus. In doing so, however, we had to give up other synergies that already existed. Each type has its advantages as well as disadvantages."* This illustrates the need for context-specific consideration, as it must be clear what the precise goal of the anticipated change shall be from a strategic business perspective. The example represents a best practice for this specific context and shows that this change in the structure as well as in the process is subsequently reflected in a significantly changed product structure. The change in the organizational structure and the resulting changes in communication flows show that Melvin Conway was not mistaken in his original postulate that the product structure should be reflected in the communication structures of the team members. This practical example can thus be seen as a further validation example of Conway's Law.

3 Establishing an appropriate corporate culture

The organizational theorist Ralph Stacey, who gave his name to the Stacey matrix in chapter 4.1.1, describes an organization as a "body of people" (Stacey, 1996). This composition of different individuals holds enormous potential, as it opens up the "space for novelty" that arises when individuals are pushed towards the "edge of chaos". Conversely, an organization and its guidelines can also have negative effects on the way individuals work, which is why the saying "culture eats strategy for breakfast" is well-known, especially in the agile context. Since a corporate culture is very multi-faceted, as the individuals themselves shape it, three aspects should be emphasized here in particular, which have often been addressed in discussions with practitioners based on the experiences of recent years: The *role of the manager*, the *willingness to change* and the necessary accompanying support of management, as well as a corresponding *open error culture*.

Role of the manager

In addition to a change in the team's composition, the manager's role is another key aspect that needs to change, from a prescriptive and controlling position to a so-called "servant leader". In this function, one of his/her tasks is to support the team members as best as possible in what they need, instead of dictating and imposing the work on them. Thus, servant leadership represents a form of work support as opposed to the delegation of work and control of the fulfillment of such. The characteristics that should be ascribed to a leader in an agile context are detailed in Atzberger et al. (2020a). Here, characteristics of servant leadership are presented based on the decision-making freedoms currently granted to teams by their superiors. For this reason, reference is made here to (Atzberger et al., 2020a, p. 16).

Willingness to change

The willingness of an organization to change represents the second necessary facet. This is often referred to as the "ivory tower" or "prince problem" (Atzberger and Paetzold, 2019) and implies that management demands a certain change in strategy, such as the introduction of agile working methods, but they do not involve themselves in the change process. Therefore, for the practitioners on the operational level, the directives on the management level often seem like directives from the princes (who are located on the ivory tower). An example of this is *"the demand for transparency in the work processes of the teams, whereas there is no transparency from management in the strategic decisions for the teams. This often gives the teams the impression of micro-management and quickly turns into rejection as the people who set the guidelines do not want to be measured by these"* (Interviewee O).

The introduction of agility in the active pursuit and implementation of agile development is also a change management process. In order to have a lasting impact on the corporate culture, it is therefore essential to create a willingness to change. Successful implementation depends heavily on the companies' willingness to change, as Cram and Newell show. Cram has conducted long-term studies on this and classifies companies into three categories: there are *"crusaders, who exclusively adopt agile [process models] in a pure form, tailors, who integrate agile and classical approaches to fit their specific circumstances; and dabblers who employ a few ceremonial agile activities alongside a classical approach"* (Cram and Newell, 2016, p. 154). Current studies also show that knowledge transfer and active and consistent change management are key success factors in introducing agile working methods (Atzberger et al., 2020a).

Open error culture

Although the open error culture represents one element of the AiM model, it is of such central importance that it is mentioned again here. In recent literature, this is referred to at times, e.g., in Ehrlenspiel and Meerkamm: *"In the sense of the Japanese attitude [...] one should be happy to discover a mistake or a weakness because every discovered mistake opens up the opportunity to do something better. (According to Prof. G. Niemann: "A mistake made must be worthwhile"). This, and not the obsessive search for culprits, is typical of creative and cooperative people and companies with an 'error culture'. This also includes admitting one's own mistakes instead of shifting them onto employees"* (Ehrlenspiel and Meerkamm, 2017, p. 168). Interviewee C again underlines its necessity: *"If I discard a concept at a relatively early stage, then it must be positive and also be perceived positively. It is simply impossible that an ideal project is*

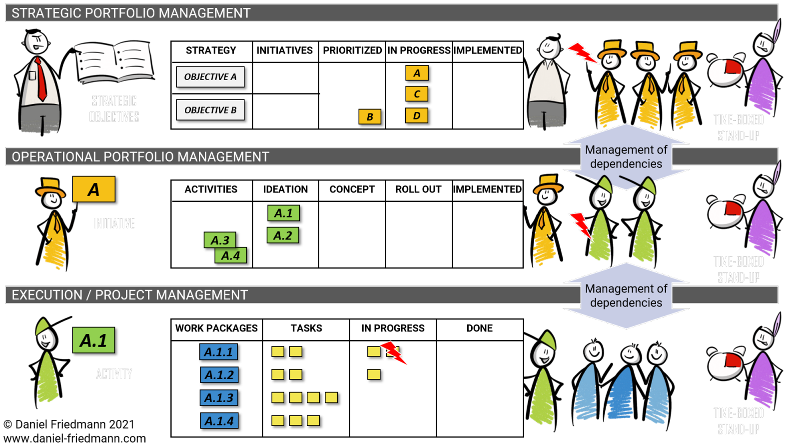


Figure 7.7
Practical implementation of agility in mechatronics. Image licensed under Common Creative and used with permission of the author.

characterized by zero mistakes, which just went through like this without any corrections. If that was the goal, we would be on the wrong track these days. This open error culture is also very, very important for development, especially in mechatronics." Nevertheless, although an open error culture exists on paper today, it is not yet consistently practiced in industrial practice, which was confirmed by the other interviewees and represents another central lever from a cultural perspective.

Implementation of agility in practice

Since all practical examples are derived from the same company, figure 7.7 summarizes the findings from all three sections in one final example from practice. The illustration corresponds to the customer delivery phase (cf. figure 7.4), as work is done at the operational level using Kanban. From a structural point of view, three hierarchical levels are presented here, as shown in figure 7.6. Thus, figure 7.7 is the summarizing representation of how a successful agile transformation can succeed in practice. On the left, the strategic objectives are broken down into initiatives, and these are concretized into activities on the second level. These are then handled by individual teams accordingly. If problems arise, they are discussed in regular, time-boxed meetings, and solutions are come up with, as shown on the right. If a problem cannot be solved directly, it is discussed in the regular meetings one level higher, which occur at the same intervals as the regular team meetings. In this way, problems are discussed openly, and at the same time, a solution is brought about as soon as possible. In order to ensure the ability to act quickly, these regular meetings take place twice a week in the company from the practical example, at all levels, and are coordinated from the lowest to the highest level (timewise).

The mixture of instruments from both Kanban (Kanban board) and Scrum (Time-boxed Stand-up) in an overall mechatronic-oriented procedure is well illustrated in this practical example. Furthermore, on the macro-level it follows the modified Stage-Gate variant of figure 7.4, i.e., the gates between the phases are (in the company's case from the practical example) adapted innovation maturity gates. Hence, this example serves quite well for illustrating what an "own process model" can look like and how it can be implemented in the context of mechatronics.

Application in practice

This chapter summarizes the most important aspects of how an agile transformation can succeed in the context of mechatronics. However, this result is the outcome of a long learning process that has taken 1.5 years and is not yet completed. Moreover, it is a potential approach tailored to the needs of the company and its products. A forthright adoption of this process model as a best practice example for mechatronics will not be expedient here either, as the underlying values and strategic objectives must first be reconciled. Nonetheless, this example shows that the application of agile development in mechatronics is feasible and can serve as an aid for other companies that are currently in the process of an agile transformation or are planning such a transformation.

8

Conclusion

"Each slight variation, if useful, will prevail." - Interviewee A, referring to Charles Darwin[1]

Summary of work Based on Darwin's quote, which is more valid today than ever before, companies must focus on the value contribution or what they have identified as value contribution and thus useful to prevail. Before an outlook for agility in mechatronics based on the interviewees' assessment is given as a conclusion of this thesis, three key takeaways of this thesis are summarized, and the limitations of this thesis concerning the result, the AiM model, are pointed out.

Change is omnipresent

Klaus Fröhlich, a former member of the board of management at BMW, mentioned in 2015 that *"the automobile will change more in the next decade than in the last fifty years"* (Böhmer, 2018). While BMW was a pioneer with its electrification strategy in Germany at that time, things look very different in 2021. Today, Tesla is the benchmark for electric vehicles (in Europe), with many other companies increasing their efforts in this regard. Concerning the development of an extremely powerful powertrain including intelligent battery technology, Tesla has also managed to implement the possibilities of digitalization directly into the mechatronic system of the car, raising the bar even higher for their competitors. These new technological possibilities were the reason for the revision of VDI 2206 (2020). Therefore, this increasing change is accompanied by the need to design processes so that they can keep up with the pace of this constant change. From that perspective, a shift towards agile ways of working is inevitable.

For this reason, this work aims to present the necessary normative elements for an application of agile working methods in the context of mechatronics. The work focuses on the design of the agile development process. In order to grasp the topic holistically, the AiM model which has been developed in this thesis is represented in the form of a tetrahedron. Its four cornerstones *process*, *product*, *person*, and *customer* characterize the environment of product development. The AiM model reflects the perspective of agility in mechatronics development. This per-

[1] Original quote by Charles Darwin: *"I have called this principle, by which each slight variation, if useful, is preserved, by the term of natural selection, in order to mark its relation to man's power of selection"* (Darwin 1859).

spective is compared to agile software development and classical (plan-driven) mechatronics development.

There are certain similarities as well as differences that must be taken into account in each perspective. A central aspect in mechatronics development is the complexity of the systems to be developed, which entails certain dependencies due to the physicality of the product to be created, which cannot be completely resolved (Schmidt, 2019). For this reason, according to Böhmer, *"hybrid models [...] are necessary for the development of complex systems"*, which is supported by the current and forecast data in recent surveys (Atzberger et al., 2020a).

3 Takeaways

In addition to the direct result of this work in the form of the AiM model, the following three takeaways are summarized briefly, which are relevant in the context of agile development of mechatronic systems.

1 Classical procedures are becoming increasingly agile

In the context of this work, as well as in conversations with other practitioners, it is evident that new ways of working (similar to new technologies in the Kano model) also become state of the art over time. This is recognizable for agile ways of working as well according to the participants of interview series №1. Although initial tendencies are discernible, there are nevertheless aspects that are already common sense in theory, but are often not lived in practice, such as an open error culture or working in cross-functional, interdisciplinary teams (Böhmer, 2018; Nicklas et al., 2021).

2 An organization is a "body of people"

As organization theorist Ralph Stacey once wrote, an organization is the interaction of different people towards a common goal, which is why he called an organization a "body of people". In order to influence the interaction of individuals, each person has to be influenced. In order to bring a new concept like agility closer to the people, their mindset has to be affected. Interviewee D mentioned *"imprint, attitude, action"* when he tries to convey an agile mindset to people in his role as an Agile Coach: *"Of course, you have to talk about agility with managers, decision-makers, not just with the employee [...], talk about value work. However, you also have to actually go into projects to make it tangible and perceptible at some point. It's like riding a bike, I can explain it, but you actually have to sit on the bike to feel it. It's the same with agility; you have to see how transparency works, you have to see how self-organization works, and only by experiencing this and drawing your experiences from it, only then does your attitude change in this regard: imprint, attitude, action. I am imprinted. This gives rise to an attitude, and actions result from the attitude. I have to change the imprint in order to change an action ultimately, so to speak. And that only works through experience."* Experiencing agile ways of working is, therefore, a key success factor.

3 Transparency is key

Another aspect that is already emerging from the advantage of agile working methods is a significant improvement in transparency in a project. This results in the disclosure of the often invisible data and information flows, which is the actual benefit of cooperation between the people. Interviewee G mentions in this context the use of a project plan in the form of a Gantt chart, which is now set up in a designated project room where regular review meetings take place in their agile project. This disclosure of information previously located in specific documents to which not every team member had access to is an immediate effect of transparency now, making the difference in an agile project.

Another interesting approach beyond this thesis's scope would be to apply the Design for Values approach in a much wider context. Kantelberg considers the maximization of productivity as the overarching goal that results from the effective and efficient development of technical products. Since productivity thus symbolizes the overarching value, all further actions are thus aligned with it. An interesting thought experiment would be to replace productivity as the overriding goal of economic activity with another one, for example sustainability. This would be possible using value-oriented action with the help of the Design for Values approach. Such a value-based "model project" is currently being implemented in the city of Gothenburg, where the city center will become climate-neutral by 2030, employing a concrete set of measures linked to ensuring sustainability to achieve this aim.

Limitations The contribution of the AiM model fulfills the initial claim, but it is limited in some aspects, as shown in table 1.1.
From a methodological point of view, it is not a method in its own right, nor is it an extension of previous agile methodologies, as Schmidt has called for. Due to its normative character, it cannot be applied directly, as it must first be tailored to one's own context and then operationalized. Moreover, it does not represent any changes to the Manifesto for Agile Software Development but functions as a standalone model for the context of mechatronics. It aims to present an image of industrial practice based on generic normative elements intended to provide a new reference.
From a company-related perspective, the model is tailored to the individuals and their interactions with other people in their immediate environment. Hence, it does not provide support for the problem areas of scaling, ecosystem, or decentralized development. It focuses on strategic goals rather than operational teamwork, so it cannot serve as a best practice for immediate problem-solving. Rather, it provides answers to why certain aspects do not work when viewed from a higher-level perspective. Furthermore, it is only designed for the German-speaking region, as only the values relevant in that region have been considered. The focus is any technical product that represents a mechatronic or cyber-physical product or system in terms of product type. Associated services or product-service systems also fall under this description. Only for pure software products this model is rather inappropriate, as it considers the context of mechatronics in particular. Here, one can fall back on the multitude of existing process models, as shown in chapter 2.2.3.

Outlook Based on this study, this model could be used and applied in different companies so that a mechatronic-specific process model can be developed on this theoretical basis and the already existing solution approaches in the form of principles for hardware as well as agile practices, which goes beyond a hybrid model such as an Agile-Stage Gate. The combination of both models makes

sense, but a new development could optimize the existing shortcomings of this combination. In addition, a complexity-dependent agility categorization for different sectors might be meaningful so that products from plant and mechanical engineering or the automotive industry have different focal points due to their diversity compared to, e.g., consumer goods electronics or medical technology.
Another field of research in this context is scaled agile mechatronics development, which considers the coordination and communication of several teams. Furthermore, the possibilities of virtual product development for the early validation of prototypes are a topic area that is only effectively used by a small proportion (25 %) in industrial practice but holds enormous potential for overcoming physical restrictions.
As far as the future application of agile working methods in the next decade is concerned, the participants in the first series of interviews agree that agility is more than just a trend whose hype will soon be over. In particular, the change in interaction amongst people and the experience of positive effects such as increased commitment between team members, increased communication, and an increase in transparency lead to a lasting positive inclination towards agility. From this perspective, the interview participants foresee agility as a general approach in the industry that will prevail in the next ten years. Nevertheless, there is still sufficient work for academia and practice, as *"we haven't cracked the nut on some issues yet"*, as interviewee B mentions.

Bibliography

Abeln, O. (1990). *Die CA...-Techniken in der industriellen Praxis: Handbuch der computergestützten Ingenieur-Methoden*. Hanser, München. OCLC: 75163739.

Abramovici, M., Herzog, O., and der Technikwissenschaften, D. A., editors (2016). *Engineering im Umfeld von Industrie 4.0: Einschätzungen und Handlungsbedarf*. acatech Studie. Herbert Utz Verlag, München. OCLC: 962553752.

Ackoff, R. L. (1994). Systems thinking and thinking systems. *System Dynamics Review*, 10(2-3):175–188.

Ackoff, R. L. (1999). A Lifetime of Systems Thinking. *The Systems Thinker*, 10(5):1–4.

Ahmed-Kristensen, S. and Daalhuizen, J. (2015). Pioneering the combined use of Agile and Stage-Gate Models in new product development - Cases from the manufacturig industry. In *Proceedings of IPDMC 2015*, Copenhagen.

Albers, A., Behrendt, M., Klingler, S., Reiss, N., and Bursac, N. (2017). Agile product engineering through continuous validation in PGE – Product Generation Engineering. *Design Science*.

Albers, A., Bursac, N., and Wintergerst, E. (2015). Product Generation Development – importance and challenges from a design research perspective. *New Developments in Mechanics and Mechanical Engineering*, pages 16–21.

Albers, A., Heimicke, J., Müller, J., and Spadinger, M. (2019a). Agility and its Features in Mechatronic System Development: A Systematic Literature Review. Florence.

Albers, A., Heimicke, J., Spadinger, M., Reiss, N., Breitschuh, J., Richter, T., Bursac, N., and Marthaler, F. (2019b). A systematic approach to situation-adequate mechatronic system development by ASD - Agile Systems Design. *Procedia CIRP*, 84:1015–1022.

Albers, A., Reiss, N., Bursac, N., and Breitschuh, J. (2016a). 15 Years of SPALTEN Problem Solving Methodology in Product Development. In Boks, C., editor, *Proceedings of NordDesign 2016*, pages 411–420.

Albers, A., Reiss, N., Bursac, N., and Richter, T. (2016b). iPeM – Integrated Product Engineering Model in Context of Product Generation Engineering. *Procedia CIRP*, 50:100–105.

Albers, A., Saak, M., and Burkhardt, N. (2002). Gezielte Problemlösung bei der Produktentwicklung mit Hilfe der SPALTEN-Methode. In *47. Internationales Wissenschaftliches Kolloquium*, Ilmenau.

Alqudah, M. and Razali, R. (2016). A Review of Scaling Agile Methods in Large Software Development. *International Journal on Advanced Science, Engineering and Information Technology*, 6(6):828.

Ambler, S. W. (2002). *Agile modeling: effective practices for eXtreme programming and the unified process*. J. Wiley, New York.

Ambler, S. W. (2009). The agile scaling model (ASM): Adapting agile methods for complex environments. *IBM December*, pages 1–35.

Ambler, S. W. and Lines, M. (2012). *Disciplined agile delivery: a practitioner's guide to agile software delivery in the enterprise*. IBM Press, Upper Saddle River, NJ.

Ambler, S. W. and Lines, M. (2020). *Choose your WoW! a disciplined agile delivery handbook for optimizing your way of working*. Project Management Institute, Inc, Newtown Square, Pennsylvania.

Anderson, D. J. (2010). *Kanban: successful evolutionary change for your technology business*. Blue Hole Press, Sequim, Washington. OCLC: 699604917.

Arnold, J. E. (1959). Creativity in Engineering. Technical report, Art Directors Club of New York, New York.

Astels, D. (2003). *Test-driven development: a practical guide*. The Coad series. Prentice Hall PTR, Upper Saddle River, NJ, 3rd edition. OCLC: 249138969.

Atzberger, A., Boffa, E., Montero, J., Paetzold, K., and Vajna, S. (2019a). How to Design for Joy / Satisfaction - Two Different Approaches on How to Tackle the Task. *Proceedings of the Design Society: International Conference on Engineering Design*, 1(1):3919–3928.

Atzberger, A., Gerling, C., Schrof, J., Schmidt, T. S., Weiss, S., and Paetzold, K. (2019b). Evolution of the Hype around Agile Hardware Development. In *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 1–8, Valbonne Sophia-Antipolis, France. IEEE.

Atzberger, A., Montero, J., and Paetzold, K. (2018). Characteristics of a metal additive manufacturing process for the production of spare parts. In *Proceedings of Symposium on Design for X*, pages 83–94, München.

Atzberger, A., Nicklas, S. J., Schrof, J., Weiss, S., and Paetzold, K. (2020a). Agile Entwicklung physischer Produkte: Eine Studie zum aktuellen Stand in der industriellen Praxis. Publisher: Universität der Bundeswehr München.

Atzberger, A. and Paetzold, K. (2019). Current Challenges of Agile Hardware Development: What are Still the Pain Points Nowadays? *Proceedings of the Design Society: International Conference on Engineering Design*, 1(1):2209–2218.

Atzberger, A., Wallisch, A., Nicklas, S., and Paetzold, K. (2020b). Antagonizing Ambiguity – Towards a Taxonomy for Agile Development. *Procedia CIRP*, 91:464–471.

Beck, K. (1999). *Extreme programming eXplained: embrace change*. Addison-Wesley, Reading, MA.

Beck, K. (2002). *Test-driven development: by example*. The Addison-Wesley signature series. Addison-Wesley, Boston.

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

Bender, B. and Gericke, K., editors (2021). *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bennett, N. and Lemoine, G. J. (2014). What a difference a word makes: Understanding threats to performance in a VUCA world. *Business Horizons*, 57(3):311–317.

Berger, C. and Eklund, U. (2015). Expectations and Challenges from Scaling Agile in Mechatronics-Driven Companies – A Comparative Case Study. In Lassenius, C., Dingsøyr, T., and Paasivaara, M., editors, *Agile Processes in Software Engineering and Extreme Programming*, volume 212, pages 15–26. Springer International Publishing, Cham.

Blanchard, B. S. and Fabrycky, W. J. (2014). *Systems engineering and analysis*. Always learning. Pearson, Essex, 5th edition. OCLC: 881697166.

Bleicher, K. (1991). *Das Konzept integriertes Management*. St. Galler Management-Konzept. Campus, Frankfurt ; New York.

Blessing, L. T. and Chakrabarti, A. (2009). *DRM, a Design Research Methodology*. Springer London, London.

Bowker, G. C. and Star, S. L. (2000). *Sorting things out: classification and its consequences*. Inside technology. The MIT Press, Cambridge, Massachusetts; London, England. OCLC: 247920023.

Boyd, J. (1995). The Essence of Winning and Losing.

Breitschuh, J., Albers, A., Seyb, P., Hohler, S., Benz, J., Reiss, N., and Bursac, N. (2018). Scaling agile practices on different time scopes for complex problem-solving. In *Proceedings of NordDesign 2018*, Linköping.

Bullinger, H.-J. and Warschat, J., editors (1996). *Concurrent Simultaneous Engineering Systems*. Springer London, London.

Böhmer, A. (2018). *When digital meets physical - Agile Innovation of Mechatronic Systems*. PhD thesis, Technische Universität München, München.

Böhmer, A. I., Beckmann, A., and Lindemann, U. (2015). Open Innovation Ecosystem - Makerspaces within an Agile Innovation Process. In *ISPIM Innovation Conference*, Brisbane.

Christopher, M. and Towill, D. (2001). An integrated model for the design of agile supply chains. *International Journal of Physical Distribution & Logistics Management*, 31(4):235–246.

Coad, P., Lefebvre, E., and De Luca, J. (1999). *Java modeling in color with UML: enterprise components and process*. Prentice Hall PTR, Upper Saddle River, NJ.

Cockburn, A. (1998). *Surviving object-oriented projects: a manager's guide*. The Addison-Wesley object technology series. Addison Wesley, Reading.

Cockburn, A. (2002). *Agile software development*. Agile software development series. Addison-Wesley, Boston. OCLC: ocm48757821.

Conboy, K. (2009). Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information Systems Research*, 20(3):329–354.

Conboy, K., Coyle, S., Wang, X., and Pikkarainen, M. (2011). People over Process: Key Challenges in Agile Development. *IEEE Software*, 28(4):48–57.

Conforto, E. C. and Amaral, D. C. (2016). Agile project management and stage-gate model—A hybrid framework for technology-based companies. *Journal of Engineering and Technology Management*, 40:1–14.

Conforto, E. C., Salum, F., Amaral, D. C., da Silva, S. L., and de Almeida, L. F. M. (2014). Can Agile Project Management be Adopted by Industries Other than Software Development? *Project Management Journal*, 45(3):21–34.

Conway, M. (1968). How do comittees invent? *Datamation magazine*, F.D. Thompson Publications:28–31.

Cooper, R. G. (1986). *Winning at new products*. Holt, Rinehart and Winston of Canada, Toronto. OCLC: 503023643.

Cooper, R. G. (2016). Agile–Stage-Gate Hybrids: The Next Stage for Product Development. *Research-Technology Management*, 59(1):21–29.

Cooper, R. G. and Sommer, A. F. (2018). Agile–Stage-Gate for Manufacturers: Changing the Way New Products Are Developed. *Research-Technology Management*, 61(2):17–26.

Cooper, R. G. and Sommer, A. F. (2020). New-Product Portfolio Management with Agile: Challenges and Solutions for Manufacturers Using Agile Development Methods. *Research-Technology Management*, 63(1):29–38.

Cram, W. A. (2019). Agile Development in Practice: Lessons from the Trenches. *Information Systems Management*, 36(1):2–14.

Cram, W. A. and Newell, S. (2016). Mindful revolution or mindless trend? Examining agile development as a management fashion. *European Journal of Information Systems*, 25(2):154–169.

Creswell, J. W. and Plano Clark, V. L. (2007). *Designing and conducting mixed methods research*. SAGE Publications, Thousand Oaks, Calif. OCLC: ocm65399935.

Creswell, J. W. and Plano Clark, V. L. (2011). *Designing and conducting mixed methods research*. SAGE Publications, Los Angeles, 2nd edition. OCLC: ocn558676948.

Czichos, H. (2019). Technische Systeme. In *Mechatronik*, pages 9–25. Springer Fachmedien Wiesbaden, Wiesbaden.

de Weck, O., Eckert, C., and Clarkson, J. (2007). A classification of uncertainty for early product and system design. In *Proceedings of the International Conference on Engineering Design*, Paris.

Debois, P. (2009). DevOps Days Ghent.

Delft Design for Values Institute (2019). Value Operationalization; From Abstract & General Values to Concrete & Specific Design Requirements.

Dikert, K., Paasivaara, M., and Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, 119:87–108.

Disciplined Agile Consortium (2017). The Disciplined Agile Enterprise (DAE).

Douglass, B. P. (2016). What are agile methods and why should I care? In *Agile systems engineering*, pages 41–84. Morgan Kaufmann, Waltham. OCLC: 922528510.

Dove, R. (1999). Knowledge management, response ability, and the agile enterprise. *Journal of Knowledge Management*, 3(1):18–35.

Dörner, D. (1976). *Problemlösen als Informationsverarbeitung*. PhD thesis, Universität Bamberg, Bamberg.

Earl, C., Johnson, J., and Eckert, C. (2005). Complexity. In Clarkson, J. and Eckert, C., editors, *Design process improvement: a review of current practice*. Springer Science & Business Media.

Eckert, C., Isaksson, O., Hallstedt, S., Malmqvist, J., Öhrwall Rönnbäck, A., and Panarotto, M. (2019). Industry Trends to 2040. *Proceedings of the International Conference on Engineering Design*, 1(1):2121–2128.

Ehrlenspiel, K. and Meerkamm, H. (2017). *Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit*. München Wien, 6th edition. OCLC: 960903430.

Eigner, M. (2014). Einleitung – Modellbasierte Virtuelle Produktentwicklung. In Eigner, M., Roubanov, D., and Zafirov, R., editors, *Modellbasierte Virtuelle Produktentwicklung*, pages 1–13. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ellis, G. (2016). *Project management in product development: leadership skills and management techniques to deliver great products*. Elsevier, Amsterdam.

Epping, T. (2011). *Kanban für die Softwareentwicklung*. Informatik im Fokus. Springer Berlin Heidelberg, Berlin, Heidelberg.

Erickson, J., Lyytinen, K., and Siau, K. (2005). Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research. *Journal of Database Management*, 16.

Evans, E. (2003). *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley, Boston.

Feldhusen, J. and Grote, K.-H. (2013a). Der Produktentstehungsprozess (PEP). In Feldhusen, J. and Grote, K.-H., editors, *Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung*, pages 11–24. Springer, Berlin, Heidelberg.

Feldhusen, J. and Grote, K.-H., editors (2013b). *Pahl/Beitz Konstruktionslehre*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Feldhusen, J., Löwer, M., and Bungert, F. (2009). Agile Methods for Design to Customer. pages 451–460, Stanford. Design Society.

Flick, U. (2002). *Qualitative Sozialforschung: eine Einführung*. Rowohlts Enzyklopädie. Rowohlt Taschenbuch Verlag, Hamburg, 6th edition. OCLC: ocm50868885.

Flick, U. (2017). *Qualitative Sozialforschung: eine Einführung*. Rowohlts Enzyklopädie. Rowohlt Taschenbuch Verlag, Hamburg, 8th edition. OCLC: 1003488430.

Frankena, W. K. (2017). *Ethik*. Springer Fachmedien Wiesbaden, Wiesbaden.

Freisleben, D. and Schabacker, M. (2002). Wissensbasierte Projektnavigation in der Produktentwicklung. In *Proceedings of the 20th CAD-FEM Users' Meeting 2002*, Friedrichshafen.

Fricke, E. and Schulz, A. P. (2005). Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle. *Systems Engineering*, 8(4).

Friedman, B., editor (1997). *Human values and the design of computer technology*. CSLI lecture notes. Cambridge University Press, Cambridge; New York.

Friedman, B., Kahn, P. H., and Borning, A. (2008). Value Sensitive Design and Information Systems. In Himma, K. E. and Tavani, H. T., editors, *The handbook of information and computer ethics*, pages 69–101. John Wiley & Sons, Inc.

Friedman, B. and Khan, P. H. J. (2003). Human values, ethics, and design. In Jacko, J. and Sears, A., editors, *The Human-Computer Interaction Handbook*, pages 1178–1196. Lawrence Erlbaum Associates, Mahwah.

Gandomani, T. J., Zulzalil, H., Ghani, A. A., Sultan, A. B., and Sharif, K. Y. (2014). How Human Aspects Impress Agile Software Development Transition and Adoption. *International Journal of Software Engineering and Its Applications*, 8(1):129–148.

Gausemeier, J., Dumitrescu, R., Echterfeld, J., Pfänder, T., Steffen, D., and Thielemann, F. (2018). *Innovationen für die Märkte von morgen: Strategische Planung von Produkten, Dienstleistungen und Geschäftsmodellen*. Carl Hanser Verlag GmbH & Co. KG, München.

Gausemeier, J. and Plass, C. (2014). *Zukunftsorientierte Unternehmensgestaltung: Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen*. Carl Hanser Verlag, München.

Gericke, K., Adolphy, S., Qureshi, A. J., Blessing, L., and Stark, R. (2021). Opening up Design Methodology. In Blessing, L., Qureshi, A. J., and Gericke, K., editors, *The Future of Transdisciplinary Design*, pages 3–14. Springer International Publishing, Cham.

Gericke, K. and Blessing, L. (2011). Comparisons of design methodologies and process models across disciplines: A literature review. In *Proceedings of the International Conference on Engineering Design*, pages 393–404.

Gericke, K. and Blessing, L. (2012). An analysis of design process models across disciplines. In *Proceedings of DESIGN 2012*, pages 171 – 180, Dubrovnik.

Gericke, K., Meißner, M., and Paetzold, K. (2013). Understanding the context of product development. In *Proceedings of the International Conference on Engineering Design*, pages 191–200, Seoul.

Goldkuhl, G. (2004). Design Theories in Information Systems - A Need for Multi-Grounding. *Journal of Information Technology Theory and Application*, 6(2):59–72.

Gräßler, I. and Hentze, J. (2020). The new V-Model of VDI 2206 and its validation. *at - Automatisierungstechnik*, 68(5):312–324.

Guaragni, F., Schmidt, T., and Paetzold, K. (2016). Traditional and Agile Product Development in a Hyperconnected World: Turning Weaknesses into Strengths. *Procedia CIRP*, 52:62–67.

Haberfellner, R. and de Weck, O. (2005). Agile SYSTEMS ENGINEERING versus AGILE SYSTEMS engineering. *INCOSE International Symposium*, 15(1):1449–1465.

Haberfellner, R., de Weck, O., Fricke, E., and Vössner, S. (2019). *Systems Engineering: Fundamentals and Applications*. Springer International Publishing, Cham.

Hales, C. and Gooch, S. (2004). *Managing engineering design*. Springer Verlag, London, 2nd edition. OCLC: 830962306.

Hansen, F. (1974). *Konstruktionswissenschaft - Grundlagen und Methoden*. Carl Hanser Verlag, München.

Heimicke, J., Chen, R., and Albers, A. (2020). Agile meets plan-driven - hybrid approaches in product development: A systematic literature review. *Proceedings of the Design Society: DESIGN Conference*, 1:577–586.

Heistinger, A. (2006). Qualitative Interviews - Ein Leitfaden zur Vorbereitung und Durchführung inklusive einiger theoretischer Anmerkungen.

Henderson, R. M. and Clark, K. B. (1990). Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly*, 35(1):9.

Hevner, A. R. (2007). A Three Cycle View of Design Science Research. 19(2):7.

Hevner, A. R. and Chatterjee, S. (2010). *Design research in information systems: theory and practice*. Integrated series in information systems. Springer, New York; London. OCLC: ocn471801169.

Hevner, A. R., March, S., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.

Heymann, M. (2005). *"Kunst" und Wissenschaft in der Technik des 20. Jahrhunderts: zur Geschichte der Konstruktionswissenschaft*. Chronos-Verlag, Zürich. OCLC: 181442530.

Highsmith, J. A. (2000). *Adaptive software development: a collaborative approach to managing complex systems*. Dorset House Pub, New York.

Hollenstein, S. and Rutz, T. (2004). The Crystal Family: Agile vs. klassische Methoden der Software-Entwicklung.

Holmström, H., Fitzgerald, B., Agerfalk, P., and Conchuir, E. (2006). Agile practices reduce distance in global software development. *Information Systems Management*, 23(3):7–18.

Hossain, E., Babar, M. A., and Paik, H.-Y. (2009). Using Scrum in Global Software Development: A Systematic Literature Review. In *IEEE International Conference on Global Software Engineering*, pages 175–184, Limerick, Ireland. IEEE.

Hostettler, R., Böhmer, A. I., Lindemann, U., and Knoll, A. (2017). TAF Agile Framework - Reducing uncertainty within minimum time and resources. In *International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 767–775.

Hruschka, P., Rupp, C., Starke, G., and Eckstein, J. (2009). *Agility kompakt: Tipps für erfolgreiche Systementwicklung*. kompakt-Reihe. Spektrum Akad. Verlag, Heidelberg, 2nd edition. OCLC: 426144269.

HSE (2018). Difference Between Safety and Security. Section: Health & Safety.

INCOSE (2014). Systems Engineering Vision 2025.

Isermann, R. (2008). *Mechatronische Systeme: Grundlagen*. Springer, Berlin, 2nd edition. OCLC: 214357427.

Janes, A. A. and Succi, G. (2012). The dark side of agile software development. In *Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software - Onward! '12*, Tucson, Arizona, USA. ACM Press.

Kantelberg, J. (2018). *Gestaltung agiler Entwicklungsprozesse technischer Produkte*. Dissertation, RWTH Aachen, Aachen.

Kettunen, P. and Laanti, M. (2008). Combining agile software projects and large-scale organizational agility. *Software Process: Improvement and Practice*, 13(2):183–193.

Khan, S. U., Niazi, M., and Ahmad, R. (2011). Barriers in the selection of offshore software development outsourcing vendors: An exploratory study using a systematic literature review. *Information and Software Technology*, 53(7):693–706.

Kim, G., Humble, J., Debois, P., and Willis, J. (2017). *Das DevOps-Handbuch: Teams, Tools und Infrastrukturen erfolgreich umgestalten*. O'Reilly, Heidelberg, 1st edition. OCLC: 979555777.

Klein, T. P. (2015). *Agiles Engineering im Maschinen- und Anlagenbau*. PhD thesis, Technische Universität München, München.

Kniberg, H., Cohn, M., and Sutherland, J. (2007). *Scrum and XP from the Trenches*. C4Media. OCLC: 959955241.

Kniberg, H. and Ivarsson, A. (2012). Scaling Agile @ Spotify.

Komus, A. and Kuberg, M. (2017). Status Quo Agile 2016/2017. Status Quo Agile, Koblenz.

Korthals, K. (2014). *Wertstromanalyse in der Produktentwicklung*. Ergebnisse aus der Produktionstechnik Produktionssystematik. Apprimus-Verlag, Aachen, 1st edition. OCLC: 899749756.

Kossiakoff, A., editor (2011). *Systems engineering: principles and practice*. Wiley series in systems engineering and management. Wiley, Hoboken, NJ, 2nd edition. OCLC: 741935803.

Kovács, G. and Spens, K. M. (2005). Abductive reasoning in logistics research. *International Journal of Physical Distribution & Logistics Management*, 35(2):132–144.

Krause, D. and Gebhardt, N. (2018). *Methodische Entwicklung modularer Produktfamilien: Hohe Produktvielfalt beherrschbar entwickeln*. Springer Vieweg, Berlin.

Kruchten, P. (2004). Scaling down large projects to meet the agile "sweet spot". *Rational Edge*.

Kruchten, P. (2011). Contextualizing agile software development. *Journal of Software: Evolution and Process*, 25(4):351–361.

Kuckartz, U. (2018). *Qualitative Inhaltsanalyse: Methoden, Praxis, Computerunterstützung*. Grundlagentexte Methoden. Beltz Juventa, Weinheim Basel, 4th edition. OCLC: 1032352464.

Ladas, C. (2008). *Scrumban: Essays on Kanban System for Lean Software Develoment*. Modus Cooperandi Press, Seattle. OCLC: 781683731.

Lalsing, V. (2012). People Factors in Agile Software Development and Project Management. *International Journal of Software Engineering & Applications*, 3(1):117–137.

Larman, C. and Basili, V. (2003). Iterative and Incremental Development: A Brief History. *IEEE Computer Society*, pages 47–56.

Larman, C. and Vodde, B. (2008). *Scaling lean & agile development: thinking and organizational tools for large-scale Scrum*. Addison-Wesley, Upper Saddle River, NJ. OCLC: ocn233547812.

Lawson, B. (1997). *How designers think: the design process demystified*. Architectural Press, Oxford; Boston, 3rd edition.

Leffingwell, D. (2011). *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*. The Agile software development series. Addison-Wesley, Upper Saddle River, NJ. OCLC: ocn671385444.

Lindemann, U., editor (2016). *Handbuch Produktentwicklung*. Hanser, München. OCLC: 913009431.

Line, M. B., Nordland, O., Røstad, L., and Tøndel, I. A. (2006). Safety vs. Security? In Stamatelatos, M. G. and Blackman, H. S., editors, *Proceedings of the Eighth International Conference on Probabilistic Safety Assessment & Management (PSAM)*, pages 1202–1210. ASME Press.

Link, P. (2014). Agile Methoden im Produkt-Lifecycle-Prozess – Mit agilen Methoden die Komplexität im Innovationsprozess handhaben. In Schoeneberg, K.-P., editor, *Komplexitätsmanagement in Unternehmen*, pages 65–92. Springer Fachmedien Wiesbaden, Wiesbaden.

Lévárdy, V. (2006). *Model-based Framework for the Adaptive Development of Engineering Systems*. PhD thesis, Technische Universität München, München.

Lévárdy, V. and Browning, T. R. (2009). An Adaptive Process Model to Support Product Development Project Management. *IEEE Transactions on Engineering Management*, 56(4):600–620.

Machi, L. A. and McEvoy, B. T. (2016). *The literature review: six steps to success*. Corwin, Thousand Oaks, California, 3rd edition.

Maier, A. and Störrle, H. (2011). What are characteristics of engineering design processes? In *Proceedings of the International Conference on Engineering Design*, pages 188–198, Kopenhagen.

Martin, J. (1991). *Rapid application development*. Macmillan Pub. Co. ; Collier Macmillan Canada, New York : Toronto.

McCracken, D. and Jackson, M. (1982). Life cycle concept considered harmful. *ACM SIFSOFT Software Engineering Notes*, 7(2):23–32.

McMahon, P. (2006). Lessons learned using agile methods on large defense contracts. *Systems and Software Technology Conference (SSTC)*, 19(5).

Meerkamm, H. and Paetzold, K. (2009). Integrierte Produktentwicklung komplexer technischer Systeme. Skriptum zur Vorlesung Konstruktionstechnik.

Mehlstäubl, J., Atzberger, A., and Paetzold, K. (2020). General Approach to support Modelling of Data and Information Flows in Product Development. In *Proceedings of NordDesign 2020*. The Design Society.

Mettler, T., Eurich, M., and Winter, R. (2014). On the Use of Experiments in Design Science Research: A Proposition of an Evaluation Framework. *Communications of the Association for Information Systems*.

Montero, J., Atzberger, A., Bleckmann, M., Holtmannspoetter, J., and Paetzold, K. (2019a). Enhancing the Additive Manufacturing process for spare parts by applying Agile Hardware Development principles. In *IEEE 10th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT)*, pages 109–116, Cape Town, South Africa.

Montero, J., Atzberger, A., Schmidt, T. S., Bleckmann, M., Holtmannspötter, J., and Paetzold, K. (2020). Anwendung agiler Entwicklungsprinzipien für die Herstellung von Ersatzteilen mit additiven Fertigungsverfahren. In Lachmayer, R., Lippert, R. B., and Kaierle, S., editors, *Konstruktion für die Additive Fertigung 2018*, pages 209–226. Springer Berlin Heidelberg, Berlin, Heidelberg.

Montero, J., Weber, S., Bleckmann, M., Atzberger, A., Wirths, L., and Paetzold, K. (2019b). Spare part production in remote locations through Additive Manufacturing enhanced by agile development principles. pages 1–8, Valbonne Sophia-Antipolis, France. IEEE.

Nattermann, R. and Anderl, R. (2013). The W-Model – Using Systems Engineering for Adaptronics. *Procedia Computer Science*, 16:937–946.

Nerur, S. and Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM*, 50(3):79–83.

Nerur, S., Mahapatra, R., and Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5):72–78.

Newman, D. (2002). The Design Squiggle.

Nicklas, S. J., Atzberger, A., Briede-Westermeyer, J. C., and Paetzold, K. (2020). The user-driven minimum feasible product - Towards a novel approach on user integration. *Proceedings of the Design Society: DESIGN Conference*, 1:1495–1504.

Nicklas, S. J., Michalides, M., Atzberger, A., Weiss, S., and Paetzold, K. (2021). Agile Entwicklung physischer Produkte: Eine Studie zum aktuellen Stand in der industriellen Praxis während der COVID-19-Pandemie. Publisher: Universität der Bundeswehr München.

Nissenbaum, H. (2001). How computer systems embody values. *Computer*, 34(3):118–120.

Nord, T. (2016). Was ist der Unterschied zwischen Safety und Security.

North, D. (2006). Behaviour Modification - The evolution of behavior-driven development. *Better Software Magazine*, (3).

Oestereich, B. and Weiss, C. (2008). Agiles Projektmanagement. *HMD Praxis der Wirtschaftsinformatik*, 45(2):18–26.

Ohno, T. (1988). *Toyota production system: beyond large-scale production*. Productivity Press, Cambridge.

Ottosson, S. (2019). *Developing and Managing Innovation in a Fast Changing and Complex World*. Springer International Publishing, Cham.

Ovesen, N. (2012). *The challenges of becoming agile*. PhD thesis, Aalborg University.

Paetzold, K. (2017). Product and Systems Engineering/CA* Tool Chains. In Biffl, S., Lüder, A., and Gerhard, D., editors, *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*, pages 27–62. Springer International Publishing, Cham.

Paetzold, K. (2020). Skriptum zur Vorlesung: Produkt- und Innovationsmanagement.

Palmer, S. R. and Felsing, J. M. (2002). *A practical guide to feature-driven development*. The Coad series. Prentice Hall PTR, Upper Saddle River, NJ.

Patzak, G. (1982). *Systemtechnik - Planung komplexer innovativer Systeme: Grundlagen, Methoden, Techniken*. OCLC: 913684247.

Poppendieck, M. and Poppendieck, T. (2003). *Lean software development: an agile toolkit*. The agile software development series. Addison-Wesley, Boston. OCLC: 844960350.

Punkka, T. (2012). Agile Hardware and Co-Design. In *Proceesings of Embedded Systems Conference 2012*, pages 1–8, Boston.

Ries, E. (2011). *The lean startup: how today's entrepreneurs use continuous innovation to create radically successful businesses*. Crown Business, New York, 1st edition. OCLC: ocn693809631.

Rigby, D. K., Sutherland, J., and Takeuchi, H. (2016). Embracing Agile. *Harvard Business Review*, pages 40–50.

Ronkainen, J. and Abrahamsson, P. (2003). Software Development under Stringent Hardware Constraints: Do Agile Methods Have a Chance? In Goos, G., Hartmanis, J., van Leeuwen, J., Marchesi, M., and Succi, G., editors, *Extreme Programming and Agile Processes in Software Engineering*, volume 2675, pages 73–79. Springer Berlin Heidelberg, Berlin, Heidelberg.

Rowe, P. G. (1987). *Design thinking*. MIT Press, Cambridge.

Royce, W. (1970). Managing the development of large software systems. In *Proceedings of IEEE WESCON*, volume 26, pages 328–338, Los Angeles.

Scaled Agile, Inc. (2020). SAFe 5.0 Framework.

Schmidt, T., Chahin, A., Kössler, J., and Paetzold, K. (2017). Agile development and the constraints of physicality: A network theory-based cause-and-effect analysis. In *Proceedings of the 21st International Conference on Engineering Design (ICED17)*, volume 4, pages 199–208, Vancouver, Canada.

Schmidt, T. S. (2019). *Towards a Method for Agile Development in Mechatronics*. Dissertation, Universität der Bundeswehr München, Neubiberg.

Schmidt, T. S., Atzberger, A., Gerling, C., Schrof, J., Weiss, S., and Paetzold, K. (2019). Agile Development of Physical Products - An Empirical Study about Potentials, Transition and Applicability. Technical report, Universität der Bundeswehr München, München.

Schmidt, T. S., Guaragni, F., and Paetzold, K. (2016). Demerging agility, leanness, flexibility, reconfigurability and changeability: Towards a clear distinction. In *2016 International Conference on Engineering, Technology and Innovation/IEEE lnternational Technology Management Conference (ICE/ITMC)*, pages 1–11, Trondheim, Norway. IEEE.

Schmidt, T. S., Weiss, S., and Paetzold, K. (2018a). Expected vs. real effects of agile development of physical products: Apportioning the hype. In *Proceedings of DESIGN 2018*, pages 2121–2132.

Schmidt, T. S., Weiss, S., Paetzold, K., and Institut für Technische Produktentwicklung (2018b). Agile Development of Physical Products: an Empirical Study about Motivations, Potentials and Applicability. Technical report, Universität der Bundeswehr München, München. OCLC: 1032690591.

Schrof, J., Schmidt, T., and Paetzold, K. (2018). Eignungsanalyse agiler Prinzipien für die Entwicklung physischer Produkte. In *Proceedings of Symposium on Design for X*, pages 61–72, München.

Schrof, J. I., Atzberger, A., Papoutsis, E., and Paetzold, K. (2019). Potential of Technological Enablement for Agile Automotive Product Development. In *IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 1–8, Valbonne Sophia-Antipolis, France. IEEE.

Schuh, G., Diels, F., Ortlieb, C., and Schröder, S. (2017). Agile Produktentwicklung. In Brecher, C., Klocke, F., Schmitt, R., and Schuh, G., editors, *Internet of Production für agile Unternehmen: AWK Aachener Werkzeugmaschinen-Kolloquium 2017*, pages 29–51. Apprimus Verlag, Aachen, 1st edition. OCLC: 1005293759.

Schuh, G., Dölle, C., and Schlesser, S. (2018). Agile Prototyping for technical systems - Towards an adaption of the Minimum Viable Product principle.

Schuh, G., Müller, J., and Rauhut, M. (2012). Gestaltung von Produktentwicklungsprozessen. In Schuh, G., editor, *Innovationsmanagement: Handbuch Produktion und Management 3*, VDI-Buch, pages 161–247. Springer, Berlin, Heidelberg.

Schumpeter, J. A. (1939). *Business Cycles: a theoretical, historical, and statistical analysis of the capitalist process*. McGraw-Hill Book Company, Inc., New York.

Schwaber, K. and Beedle, M. (2001). *Agile software development with Scrum*. Series in agile software development. Pearson/Prentice Hall, Upper Saddle River, NJ. OCLC: 845452162.

Schwaber, K. and Scrum.org (2015). Nexus Guide.

Schwaber, K. and Sutherland, J. (2017). The Scrum Guide 2017.

Schwaber, K. and Sutherland, J. (2020). The Scrum Guide 2020.

Sharifi, H. and Zhang, Z. (2001). Agile manufacturing in practice - Application of a methodology. *International Journal of Operations & Production Management*, 21(5/6):772–794.

Smith, P. G. (2007). *Flexible Product Development: Building Agility for Changing Markets*. Jossey-Bass, San Fracisco.

Socha, D. and Skip, W. (2006). Is Designing Software Different From Designing Other Things? *International Journal of Engineering Education*, 22(3):540–550.

Sommer, A. F., Dukovska-Popovska, I., and Steger-Jensen, K. (2014). Barriers towards integrated product development — Challenges from a holistic project management perspective. *International Journal of Project Management*, 32(6):970–982.

Sommer, A. F., Hedegaard, C., Dukovska-Popovska, I., and Steger-Jensen, K. (2015). Improved Product Development Performance through Agile/Stage-Gate Hybrids: The Next-Generation Stage-Gate Process? *Research-Technology Management*, 58(1):34–45.

Specht, G., Beckmann, C., and Amelingmeyer, J. (2002). *F&E-Management: Kompetenz im Innovationsmanagement*. Schäffer-Poeschel, Stuttgart, 2nd edition. OCLC: 76314428.

Stacey, R. D. (1996). *Complexity and creativity in organizations*. Berrett-Koehler Publishers, San Francisco, 1st edition.

Stachowiak, H. (1973). *Allgemeine Modelltheorie*. Springer, Wien. OCLC: 884098.

Stapleton, J. (1997). *DSDM, dynamic systems development method: the method in practice*. Addison-Wesley, Harlow. OCLC: 246130816.

Stark, R. and Damerau, T. (2019). Digital Twin. In The International Academy for Production Engineering, Chatti, S., and Tolio, T., editors, *CIRP Encyclopedia of Production Engineering*, pages 1–8. Springer Berlin Heidelberg, Berlin, Heidelberg.

Stelzmann, E. S. (2011). *Agile Systems Engineering: Eine Methodik zum besseren Umgang mit Veränderungen bei der Entwicklung komplexer Systeme*. PhD thesis, Technische Universität Graz, Graz.

Stelzmann, E. S. (2012). Contextualizing Agile Systems Engineering. *IEEE Aerospace and Electronic Systems Magazine*, pages 17–22.

Succi, G. and Marchesi, M. (2001). *Extreme Programming examined*. The XP series. Addison-Wesley, Boston.

Sundén, J. (2013). Agile à la Spotify. Section: People.

Sutherland, J. (2005). Future of scrum: Parallel pipelining of sprints in complex projects. In *Agile Development Conference (ADC'05)*, pages 90–99, Denver, CO, USA. IEEE.

Sutherland, J. (2014). *Scrum: the art of doing twice the work in half the time*. Random House Business Books, London.

Sutherland, J. and Scrum Inc. (2018). Scrum@Scale Guide.

Takeuchi, H. and Nonaka, I. (1986). The new new product development game. *Harvard Business Review*, (1):137–146.

Thomas, D. (2015). Agile is Dead.

Thomas, R. M. (2003). *Blending qualitative & quantitative research methods in theses and dissertations*. Corwin Press, Thousand Oaks, California.

Thomke, S. and Reinertsen, D. (1998). Agile Product Development: Managing Development Flexibility in Uncertain Environments. *California Management Review*, 41(1):8–30.

Tseng, Y.-H. and Lin, C.-T. (2011). Enhancing enterprise agility by deploying agile drivers, capabilities and providers. *Information Sciences*, 181(17):3693–3708.

Tsourveloudis, N. C. and Valavanis, K. P. (2002). On the Measurement of Enterprise Agility. *Journal of Intelligent and Robotic Systems*, 33(3):329–342.

Ullman, D. G. (2010). *The mechanical design process*. McGraw-Hill series in mechanical engineering. McGraw-Hill Higher Education, Boston, 4th edition. OCLC: ocn244060468.

Ulrich, K. T. and Eppinger, S. D. (2016). *Product design and development*. McGraw-Hill Education, New York, NY, 6th edition.

Ulrich, P. and Hill, W. (1976). Wissenschaftstheoretische Grundlagen der Betriebswirtschaftslehre. *WiSt - Wirtschaftswissenschaftliches Studium*, 5(8):304 – 309;345 – 350.

Vajna, S. (2014). *Integrated Design Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg.

van de Poel, I. (2013). Translating Values into Design Requirements. In Michelfelder, D. P., McCarthy, N., and Goldberg, D. E., editors, *Philosophy and Engineering: Reflections on Practice, Principles and Process*, volume 15, pages 253–266. Springer Netherlands, Dordrecht.

van de Poel, I., Goldberg, D. E., and Davis, M., editors (2010). *Philosophy and engineering: an emerging agenda*. Philosophy of engineering and technology. Springer, Dordrecht ; New York. OCLC: ocn401158176.

van den Hoven, J. (2007). ICT and Value Sensitive Design. In Goujon, P., Lavelle, S., Duquenoy, P., Kimppa, K., and Laurent, V., editors, *The Information Society: Innovations, Legitimacy, Ethics and Democracy*, IFIP International Federation for Information Processing, pages 67–72. Springer, Boston. OCLC: 220233048.

van den Hoven, J. (2021). R101x - Value Sensitive Design.

van den Hoven, J., Vermaas, P. E., and van de Poel, I., editors (2015). *Handbook of Ethics, Values, and Technological Design*. Springer Netherlands, Dordrecht.

VDI 2206 (2004). VDI 2206: Entwicklungsmethodik für mechatronische Systeme. Technical report, VDI-Verlag, Düsseldorf.

VDI 2206 (2020). VDI 2206: Entwicklungsmethodik für mechatronische Systeme. Technical report, VDI-Verlag, Düsseldorf.

VDI 2221 (1993). VDI 2221: Methodik zum Entwickeln und Konstruiere technischer Systeme und Produkte. Technical report, VDI-Verlag, Düsseldorf.

VDI 2221 (2019a). VDI 2221 Blatt 1: Entwicklung technischer Produkte - Modell der Produktentwicklung. Technical report, VDI-Verlag, Düsseldorf.

VDI 2221 (2019b). VDI 2221 Blatt 2: Entwicklung technischer Produkte und Systeme - Gestaltung individueller Produktentwicklungsprozesse. Technical report, VDI-Verlag, Düsseldorf.

VersionOne (2019). 13th Annual State of Agile Report. Technical report.

VersionOne (2020). 14th Annual State of Agile Report. Technical report.

Visual paradigm (2020). What are the 10 Principles of LeSS Framework?

von Hippel, E. (1986). Lead Users: A Source of Novel Product Concepts. *Management Science*, 7(32):791–805.

Welge, M. and Friedrich, C. (2012). Integration von agilen Methoden in der Systementwicklung. In Maurer, M. and Schulze, S.-O., editors, *Tag des Systems Engineering: Komplexe Herausforderungen meistern*. Hanser, München. OCLC: 838985985.

Womack, J. P., Jones, D. T., and Roos, D. (1991). *The machine that changed the world: The Story of Lean Production*. Harper Perennial, New York, 1st edition.

Wynn, D. and Clarkson, J. (2005). Models of designing. In Clarkson, J. and Eckert, C., editors, *Design process improvement*, pages 34–59. Springer London, London.

Wynn, D. C. and Clarkson, P. J. (2018). Process models in design and development. *Research in Engineering Design*, 29(2):161–202.

Wynn, D. C. and Eckert, C. M. (2017). Perspectives on iteration in design and development. *Research in Engineering Design*, 28(2):153–184.

Zimmermann, B. (2001). Ralph Stacey's Agreement & Certainty Matrix. Technical report, Schulich School of Business, York University, Toronto.

Appendix

A

Appendix

A.1 Contributions of the author

Table A.1
Own investigations on knowledge generation in the research process at the author's institute.

Intent	Contribution	Method	Reference
Relevance			
Statistical analysis about the motivation, effects and applicability to test popular myths and evaluate the status of agile development of physical products	Status quo of agile development in mechatronics	Online survey, statistical analysis	Schmidt et al. (2019); Atzberger et al. (2020a); Nicklas et al. (2021)
Problem space			
Benefits of agile hardware development and assessment of the state of the hype around agile hardware development	Advancement of agile hardware development on Gartner Hype Cycle	Online survey	Atzberger et al. (2019b)
Combination of an agile framework and Generative Design process to circumvent constraints of physicality (CoP) and constraints of scale (CoS)	Enablement of new technologies (Generative Design) in an agile context for automotive product development	Participatory action research	Schrof et al. (2019)
Analysis of current challenges of agile hardware development	Overview of corresponding challenges and ‚sweet spot' model	Unstructured literature review, workshops	Atzberger and Paetzold (2019)

to be continued on the next page...

Intent	Contribution	Method	Reference
Comparison of terms used in agile vs traditional product development literature to antagonize ambiguity	Development of a taxonomy for agile hardware development	Structured literature review, theoretical reasoning	Atzberger et al. (2020b)
Application			
Analyzing the need and potential process improvements through agile development	Exemplary case study in application domain for agility	Case study	Atzberger et al. (2018)
Adaptation and advancement of AM spare part design process and evaluation of process enhancement by agile hardware development principles	Redesign and alignment of spare part design process	Case study, theoretical reasoning	Montero et al. (2019a)
Application of agile development principles on the AM spare part design production process using additive manufacturing	Application of agile hardware development principles on spare part design process	Case study	Montero et al. (2019b, 2020)
Comparison of an agile vs a traditional approach by means of a specific development project	Assessment and comparison of strengths and weaknesses of both approaches	Case study	Atzberger et al. (2019a)
Development of new concept to shift the hermeneutic workload from the user's to the designer's side.	Development of UD-MFP (user- driven minimal feasible product) concept	Structured literature review, theoretical reasoning	Nicklas et al. (2020)
Process			
Methodical approach for detailed and purpose-oriented modelling of data and information flows in product development	Holistic view on data and information flows	Literature review, theoretical reasoning	Mehlstäubl et al. (2020)

End of table

A.2 Assessment of practitioners

Practitioner evaluation of every normative element of the AiM model in Nicklas et al. (2021). The assessment of the individual elements by the entirety of practitioners (mean) and two sets of selected groups (experts and experienced) is displayed with the respective standard deviation for each dataset.

Table A.2
Evaluation of the individual elements of the AiM model in Nicklas et al. (2021).

№	**Element** (n = *varies*)	**Mean**	σ **Mean**	**Experts** (n = 37)	σ **Experts**	**Experienced** (n = 43)	σ **Experienced**
1	Trust (n = 99)	4.67	.72	4.89	.31	4.61	.85
2	Motivation (n = 99)	4.57	.71	4.75	.43	4.56	.83
3	Learning culture (n = 99)	4.51	.74	4.61	.54	4.54	.63
4	Short-cyclic gain in insight (n = 99)	3.91	.91	3.89	.97	4.07	.81
5	Continuous improvement (n = 99)	4.21	.8	4.14	.79	4.32	.75
6	Interdisciplinary collaboration (n = 99)	4.42	.8	4.31	.88	4.22	.95
7	Connectivity (n = 97)	3.76	1.1	3.89	1.01	3.79	1.11
8	Systems thinking (n = 98)	4.46	.8	4.61	.72	4.5	.81
9	Technical excellence (n = 98)	4.18	.79	4.25	.64	4.25	.62
10	Safety & Security (n = 94)	4.04	1.1	4.31	.82	4.18	.88

A.3 Comprehensive list of normative elements of all three perspectives

Table A.3
Values and principles of agile methodologies in agile software period.

Methodology	Value	Principle	Reference
RAD	-	Small teams Lower costs Reusable parts High quality Automated tools Lower maintenance costs Meets business needs better User involvement	Martin (1991)
Scrum	Commitment	Transparency	Schwaber and Beedle (2001)
	Focus	Inspect	Schwaber and Sutherland (2017)
	Openness	Adapt	
	Respect	Iterative and incremental development	
	Courage	Empirical process control Self-organization Collaboration Value-based prioritization Time-boxing	
DSDM	-	Active user involvement Empowerment of teams Focus on frequent delivery Fitness for business purpose Iterative and incremental Changes are reversible Requirements are baselined at a high level Testing Collaboration	Stapleton (1997)
Crystal	People-centricity	Direct communication	Cockburn (2002)
	Communication	Process simplicity	Hollenstein and Rutz (2004)
	Tolerant	Process adaptability	
	Safety	Feedback / Reflective Improvement	
	Efficiency	Focus	

to be continued on the next page...

Methodology	Value	Principle	Reference
		Automated testing	
XP	Communication Simplicity Feedback Courage Respect	Assumed simplicty Incremental change Embracing change Quality work Iterations	Beck (1999) Succi and Marchesi (2001)
FDD	-	Highly iterative People-focused Results-oriented / Incremental Inspection High quality Develop by feature	Coad et al. (1999) Palmer and Felsing (2002)
ASD	-	Iteration Concurrency Feedback Collaboration	Highsmith (2000)
TDD	-	Simple design Automated testing Continuous improvement	Beck (2002) Astels (2003)
AM	Communication Simplictiy Feedback Courage Humility Beyond motherhood and applepie	Software is your primary goal Incremental change Travel light Assume simplicity Embrace change Enabling the next effort is your second goal Model with a purpose Multiple models Quality work Maximize stakeholder involvement	Ambler (2002)
DDD	-	Focusing on core domain Learning through collaboration Creating models through exploration and experimentation Communication	Evans (2003)

to be continued on the next page...

Methodology	Value	Principle	Reference
		Understanding the applicability of a model Constantly evolving the model	
BDD	-	Close and clear communication Focus on value (stream) Keep the up-front process lean	North (2006)

End of table

Table A.4
Values and principles of agile scaling frameworks in agile scaling period.

Framework	Value	Principle	Reference
LeSS	-	LeSS is not Scrum	Larman and Vodde (2008)
		Empirical process control (Inspect and adapt) Transparency Whole-product focus Customer-centricity Continuous improvement towards perfection (feedback und increment) Systems thinking Queuing theory (managing WIP and queues) Lean thinking More with LeSS (more learning with less processes, more values with less waste, more ownership, purpose with fewer roles)	Visual paradigm (2020)
DevOps	-	Principle of flow Principle of feedback Continual learning and experimentation	Debois (2009) Kim et al. (2017)
SAFe	Alignment Bulit-in quality Transparency	Take an economic view Systems thinking Assume variability, preserve options	Leffingwell (2011) Scaled Agile (2020)
	Program execution	Incremental build with fast learning cycles	
	Continuous delivery	Base milestones on objective evaluation of working systems Visualize and limit WIP – manage queues Synchronization and cross-domain planning Unlock intrinsic motivation of people Decentralized decision-making Organize around value	
DAD	Changes / Additions to the	Delight customers	Ambler and Lines (2012)

to be continued on the next page...

Framework	Value	Principle	Reference
	Manifesto	Be awesome Context counts Be pragmatic Choice is good Optimize Flow Organize around products/ services Enterprise awareness	Disciplined Agile Consortium (2017)
Spotify	Continuous improvement Iterative development Simplicity Trust Servant leadership	-	Kniberg and Ivarsson (2012) Sundén (2013)
Nexus	Close to Scrum Focus on an integrated product	Transparency Increment	Schwaber and Scrum.org (2015)
Scrum@Scale	Transparency Inspection Adaption Openness Courage Focus Respect Commitment	Metric: Productivity Metric: Value delivery Metric: Quality Metric: Sustainability	Sutherland and Scrum Inc. (2018)

End of table

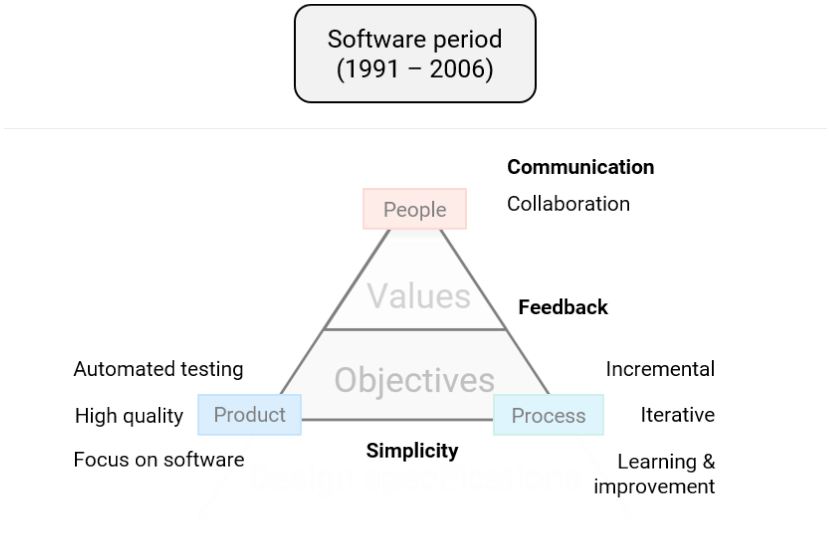
Table A.5
Normative elements of classical methodologies.

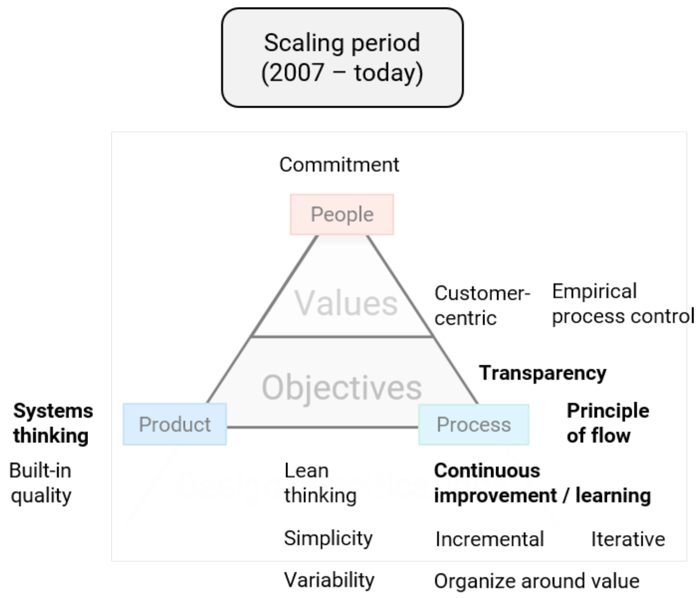
Methodology	Normative element	Reference
VDI 2221	Creativity Experience (of the individual) Nonconformity Intrinsic motivation Openness Risk-taking propensity Flexibility Wide-ranging interests Interdisciplinary procedure Problem-centered and task-relevant communication Coordination and collaboration Structured and notorious description of the procedure Function and cost optimization Technical range of requirements (product quality) Iterative procedure	VDI 2221 (2019a)
VDI 2206	Cyber security System / Operational safety Consistency, correctness & completeness (of a system) Verification & Validation Efficiency (resource-sparing) Adaptability Consistent and non-redundant communication Parallel and iterative procedure Interdisciplinary product development Inclusion of systems engineering & business model aspects Focus on requirements Assurance due to comprehensive requirements development Unambiguity, consistency, correctness & completeness of requirements Continuous planning & implementation of feature validation	VDI 2206 (2020)

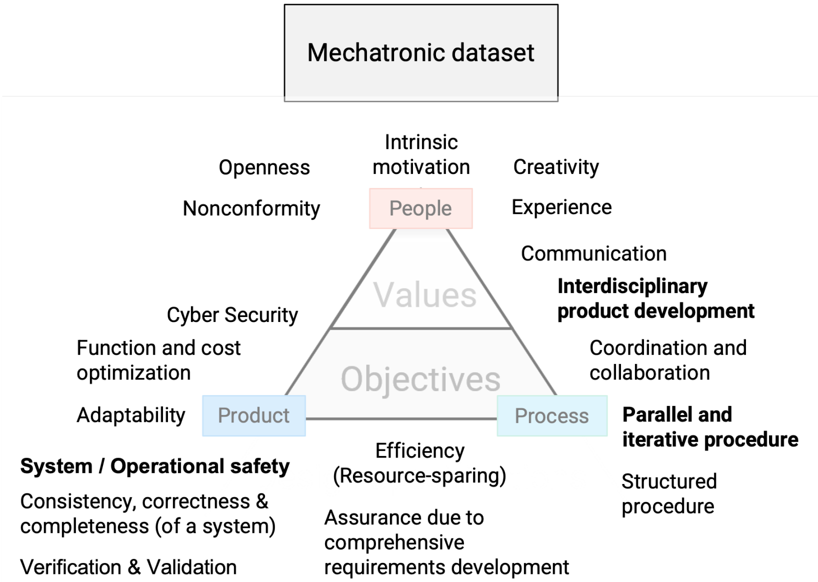
Table A.6
Values and objectives of interviewees of interview series №1.

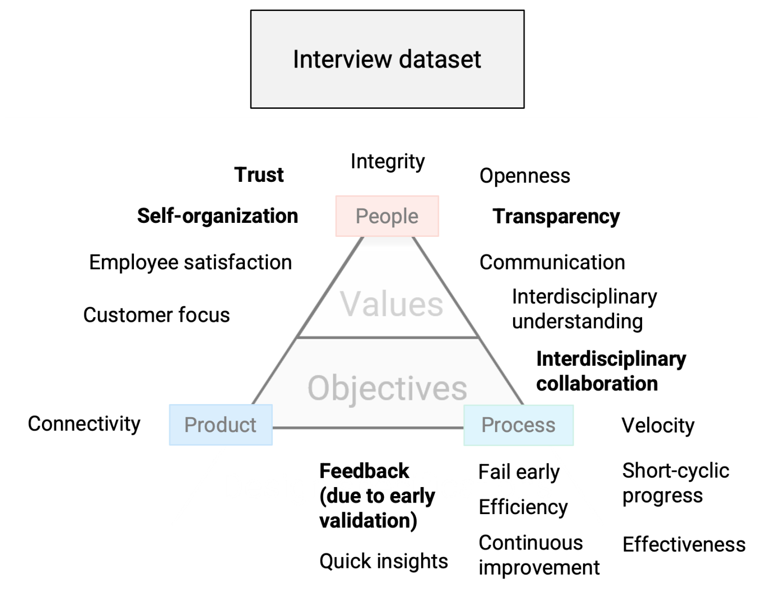
Interviews	Value	Objective
Interview series №1	Trust	Intrinsic motivation
	Commitment	Responsibility
	Courage	Respect
	Openness	Willingness to give feedback
	Self-organization	Decision-making authority
	Human-centricity	Employee satisfaction
	Integrity	Continuous exchange
	Simplicity	Cross-departmental coordination
	Flexibility	Reaction to change
	Intentionality	Coherent action
	Transparency	Mitigating efficiency loss
	Customer focus	Customer satisfaction
	Interdisciplinary communication & knowledge	Understanding of other domains' challenges
	Interdisciplinary collaboration	Shared understanding
	Open error culture (fail early)	Open communication
	Continuous improvement	Short-cyclic progress
	(Early) Feedback	Early validation
	Functional increments	Risk minimization / error prevention
	Efficiency	Faster error / problem detection
	Effectiveness	Faster intermediate results / insights
	Focus	Solution-oriented work
	Velocity	Faster delivery (time-to-market)
	Systems thinking	System integration
	Connectivity	Connected products
		Connected organizations
		Modular hardware prototypes (reduced range of functions)
		Mastering complexity
		Reduction of 'waste' in planning
		Robustness of planning
		Resource-sparing

A.4 Triangle representation of normative elements

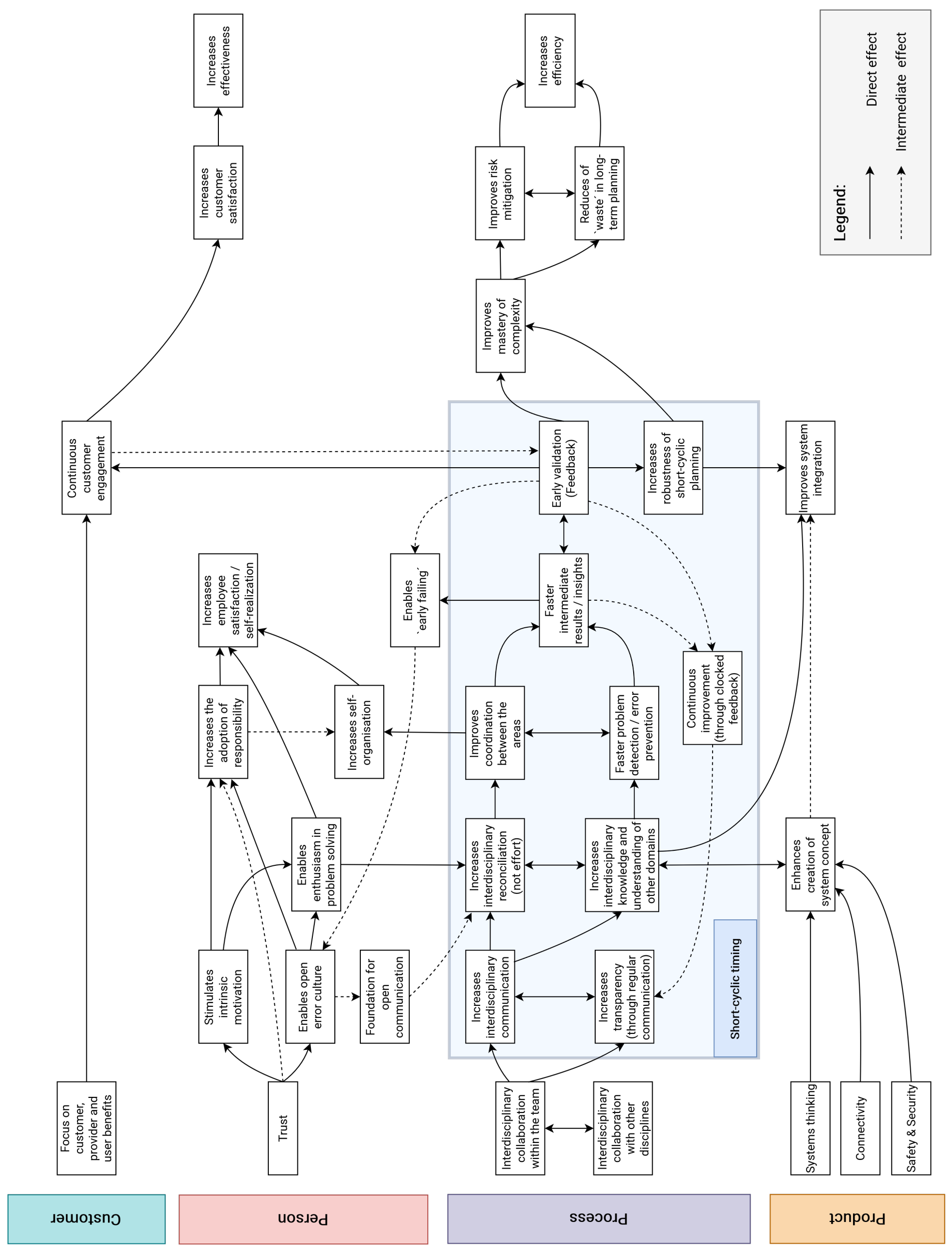





Mechatronic dataset
Openness
Intrinsic motivation
Creativity
Nonconformity
People
Experience
Communication
Values
Interdisciplinary product development
Cyber Security
Function and cost optimization
Objectives
Coordination and collaboration
Adaptability
Product
Process
Parallel and iterative procedure
System / Operational safety
Efficiency (Resource-sparing)
Consistency, correctness & completeness (of a system)
Structured procedure
Assurance due to comprehensive requirements development
Verification & Validation


Interview dataset
Trust
Integrity
Openness
Self-organization
People
Transparency
Employee satisfaction
Communication
Values
Interdisciplinary understanding
Customer focus
Interdisciplinary collaboration
Objectives
Connectivity
Product
Process
Velocity
Feedback (due to early validation)
Fail early
Short-cyclic progress
Efficiency
Continuous improvement
Effectiveness
Quick insights

A.5 Means-end relationships



A.6 Interview series № 1

Additional information about the interviewees of interview series № 1.

Name:	Position:	Domain:	Personal experience with agile process models:
Interviewee A	Agile Coach / Scrum Master	Development	8 years
Sector / Products:	**No. of employees in total / in R&D:**	**Experience with agile development overall / in mechatronics:**	**Impact of agile process models in organisation:**
Automotive Supplier (Tier 1)	240,000 / 5,000	n.a. / 2018	LOW

Name:	Position:	Domain:	Personal experience with agile process models:
Interviewee B	Project leader	Project management / Development	6 years
Sector / Products:	**No. of employees in total / in R&D:**	**Experience with agile development overall / in mechatronics:**	**Impact of agile process models in organisation:**
Plant engineering / Machine tools	14,000 / 600	2010 / 2013	HIGH

Name:	Position:	Domain:	Personal experience with agile process models:
Interviewee C	Referent development	Advance development	5 years
Sector / Products:	**No. of employees in total / in R&D:**	**Experience with agile development overall / in mechatronics:**	**Impact of agile process models in organisation:**
Consumer goods / Power tools	17,000 / 800	n.a. / 2016	MED

Name:	Position:	Domain:	Personal experience with agile process models:
Interviewee D	Agile Coach	Production	5 years
Sector / Products:	**No. of employees in total / in R&D:**	**Experience with agile development overall / in mechatronics:**	**Impact of agile process models in organisation:**
Automotive (OEM)	670,000 / 20,000	2007 / 2015	MED

Name:	Position:	Domain:	Personal experience with agile process models:
Interviewee E	Product Owner	Development	2 years
Interviewee F	Product Owner	Development	2 years

Sector / Products:	No. of employees in total:	Experience with agile development in mechatronics:	Impact of agile process models in organisation:
Engineering service provider	500	2018	HIGH

Name:	Position:	Domain:	Personal experience with agile process models:
Interviewee G	Head of project management	Development	9 years

Sector / Products:	No. of employees in total / in R&D:	Experience with agile development overall / in mechatronics:	Impact of agile process models in organisation:
Medical engineering / Emergency equipment	250 / 80	2011 / 2016	HIGH

Name:	Position:	Domain:	Personal experience with agile process models:
Interviewee H	Systems Engineer	Development	7 years

Sector / Products:	No. of employees in total / in R&D:	Experience with agile development overall / in mechatronics:	Impact of agile process models in organisation:
Agricultural machinery	11,000 / 800	2013 / 2017	MED

Name:	Position:	Domain:	Personal experience with agile process models:
Interviewee I	Agile Coach	Development	6 years

Sector / Products:	No. of employees in total / in R&D:	Experience with agile development overall / in mechatronics:	Impact of agile process models in organisation:
Automotive (OEM)	300,000 / 10,000+	n.a. / 2016	MED

Name: **Interviewee J**

Position: Research assistant

Domain: Research

Personal experience with agile process models: 4 years

Sector / Products: Academia / University

No. of employees in total: 50

Experience with agile development in mechatronics: 2017

Impact of agile process models in organisation: HIGH

Name: **Interviewee K**

Position: Research assistant

Domain: Research

Personal experience with agile process models: 2 years

Sector / Products: Academia / University

No. of employees in total: 20

Experience with agile development in mechatronics: 2018

Impact of agile process models in organisation: HIGH

Name: **Interviewee L**

Position: Managing Director

Domain: Development

Personal experience with agile process models: 12 years

Sector / Products: Consulting

No. of employees in total / in R&D: 15

Experience with agile development in mechatronics: 2008

Impact of agile process models in organisation: HIGH

A.7 Interview series № 2

Additional information about the interviewees of interview series № 2.

Name:	Position:	Domain:	Personal experience with agile process models:	Sector / Products:	No. of employees in total / in R&D:	Experience with agile development in mechatronics:	Impact of agile process models in organisation:
Interviewee L	Managing Director	Development	12 years	Consulting	15	2008	HIGH
Interviewee M	Agile Coach	Development	8 years	Consumer Electronics	2,800 / 600	2014	HIGH
Interviewee N	Head of Sales	Sales & Distribution	5 years	Plant engineering / Food processing	160 / 10	2017	HIGH
Interviewee O	Agile Coach / Expertise Owner	Advance development	6 years	Power Tools	20,000 / 5,000	Pre-2000	HIGH

Name:
Interviewee P

Position:
Corporate strategist

Domain:
Organizational Development

Personal experience with agile process models:
4 years

Sector / Products:
Plant engineering / Packaging industry

No. of employees in total / in R&D:
3,100 / 500

Experience with agile development in mechatronics:
2012

Impact of agile process models in organisation:
HIGH

Name:
Interviewee Q

Position:
Head of project management

Domain:
Development

Personal experience with agile process models:
8 years

Sector / Products:
Consumer products / Cleaning equipment

No. of employees in total / in R&D:
13,500 / 600

Experience with agile development in mechatronics:
2014

Impact of agile process models in organisation:
MED

Name:
Interviewee R

Position:
Expert Agile Transition

Domain:
Development

Personal experience with agile process models:
9 years

Sector / Products:
Plant engineering / Machine tools

No. of employees in total / in R&D:
14,000 / 3,000

Experience with agile development in mechatronics:
2009

Impact of agile process models in organisation:
HIGH