iced19

COMPARISON OF EXISTING AGILE APPROACHES IN THE CONTEXT OF MECHATRONIC SYSTEM DEVELOPMENT: POTENTIALS AND LIMITS IN IMPLEMENTATION

Heimicke, Jonas; Niever, Manuel; Zimmermann, Valentin; Klippert, Monika; Marthaler, Florian; Albers, Albert

Karlsruhe Institute for Technology (KIT), IPEK - Institute of Product Engineering

ABSTRACT

The development of mechatronic systems has always been characterized by continuous handling of uncertainties. This challenge, which is associated with dynamic changes in the development context, is increasingly met by companies in the development of physical systems with the implementation of agile approaches in their development processes. However, since established approaches have their origin in software development, they reach various limits in the context of the development of mechatronic systems, e.g. due to the physical properties of the systems. Other features, such as transparent and flexible project management or targeted and early involvement of customers and users in development processes, can also be implemented in mechatronic system development. In order to derive the potentials and limits of existing agile approaches for the context of mechatronic system development, the present paper compares existing approaches with regard to relevant factors from the context of mechatronic system development. The aim is to create a basis for the targeted development, adaptation and use of agile approaches in the field of mechatronic system development.

Keywords: Mechatronics, Integrated product development, Organisation of product development, Agile Mechatronic System Development

Contact:
Heimicke, Jonas
Karlsruher Institut für Technologie (KIT)
IPEK - Institut für Produktentwicklung
Germany
jonas.heimicke@kit.edu

1   INTRODUCTION

Dynamic market changes, latent and non-transparent customer requirements as well as the striving of companies for unique selling propositions and competitive advantages in the market are as old as the economic activities of companies (Schumpeter, 1912, 3ff.). For more than ten years, the agile design of entrepreneurial development processes has provided new answers to make development projects more robust against volatile changes in the market and the resulting uncertainties than was possible with classic processes (Smith, 2007). For example, customer requirements are now integrated earlier into the development processes, prototypes are quickly built for validation, and different iteration cycles and types are used to correlate problem and solution areas (Wynn and Eckert, 2017). In the field of mechatronic system development, too, agile approaches have found their way into the development processes, as a result of which companies have become aware of many potentials as well as various challenges that an integration of agile approaches into the development of physical products entails (Schmidt et al., 2017b). In most cases, this integration takes place through a simple transfer of agile procedures from software development to mechatronic system development or through a combination of different approaches at different project levels (e.g. phases according to VDI2221 and project implementation according to Scrum) (Schuh et al., 2018). In addition, there are numerous studies that compare established agile approaches or hybrid approaches in software development (Barlow et al., 2011). In order for an approach to support mechatronic system development in the best possible way, however, the technical orientation and the continuous integration of product and process knowledge are indispensable in order to keep the processes in the development of physical products stable. For example, companies in the automotive industry certify their processes and develop systems according to certain guidelines or assessments, such as ASPICE, which is based on ISO15504 (VDA - Verband der Automobilindustrie, 2007). Existing agile approaches, however, are only conditionally suitable for meeting such guidelines due to their different application focus. In addition, the question arises whether companies that have existed for a long time in certain markets have not already anchored many facets of agility in their culture, since companies that do not react quickly and adapt to changed market requirements cannot guarantee their ability to innovate in the long term (Tuominen et al., 2004). A very well-known example of this is Nokia, which recognised the trend towards touch operation of mobile phones, but shunned the need for major restructuring and competence restructuring and continued to operate mobile phones via keys, with well-known consequences (Laamanen et al., 2016).

For this reason, this article deals with the identification of essential factors that agile approaches in mechatronic system development must satisfy to support development teams in order to create an understanding of the needs at hand here. Based on this, different agile approaches are compared with regard to these factors.

2   METHODOLOGY

For the application-oriented development and adaptation of agile approaches in the field of mechatronic system development, an analysis of the potentials and limits of existing approaches for the intended context is necessary. Based on this, suitable elements can be combined and missing elements can be specifically generated. The resulting understanding should help researchers to develop approaches that contain suitable mechanisms for integrating a situation- and demand-oriented degree of agility in mechatronic system development processes. For this purpose, the following research questions are answered in this article:

1.   In which contexts is the concept of agility applied in product development research?
2.   Which similarities and differences of agile approaches can be determined on the basis of the comparison?
3.   Which potentials and challenges for the application in mechatronic system development result from the comparison?

In order to answer the research questions, a data mining analysis with the string *agility in product development research* is first carried out in order to identify relevant and current research trends on the subject of agility. Then suitable criteria are derived from the identified literature and selected agile approaches are compared with regard to these criteria. On this basis, an initial comparison is made to identify the limits and potentials that these approaches bring with them in mechatronic system development.

3 STATE OF THE ART

3.1 Definition of agile capabilities

Agile approaches have their origins in American software development and are understood as the opposite pole to classic, inflexible approaches such as the waterfall oriented approach (Royce, 1970) or the stage gate process (Cooper, 2015). Through e.g. decentralized decision-making processes, fast trial-and-error loops and increased collaboration in the development teams, agile approaches lead to an increased reactivity of companies to a dynamic development context. (Schmidt et al., 2018) Rebentisch et al. (2018) also take up this aspect in their definition of agility. Accordingly, agility is "the capability to discover and understand changing product requirements, and being able to quickly consider these changes while making progress in developing the product" (Rebentisch et al., 2018). This definition is to be understood independently of any domain, however, the implementation of agile approaches holds different potentials and limits depending on the respective industry (e.g. software or mechatronic system development). (Schmidt et al., 2017a) However, differences in the integration of agile approaches in development processes result in particular from different boundary conditions of the different domains software and mechatronics. Software describes all soft components that exist in the context of information technology products. The term soft describes both the intangible appearance (Broy, 2010) of the software and the possibility of simple subsequent changes. (Engesser and Claus, 1993) Software is therefore subject to fewer technical restrictions than hardware and has a very high degree of design freedom. As a result, software in development allows a very high flexibility with regard to requirements and changes. The provision of software does not involve a conventional production process, such as for mechanical products, but is provided by duplication at the end of development. (Broy, 2010) According to ISO/IEC/IEEE 24765:2010, software includes *programs, procedures, rules and associated documentation of an information processing system*. (ISO/IEC/IEEE 24765:2010, 2010)

Mechatronic systems are semi-intelligent solutions that are created through the interaction of mechanical systems with elements of electrical engineering such as sensors or microcomputers. They are thus the result of interdisciplinary development projects in the sub-disciplines of mechanical engineering, electrical engineering and information technology. This provides the software that controls the logical processes as a subproduct. (Czichos, 2015) The interaction of the various domains and the associated complexity (Schöner, 2006) influence the underlying development process in interaction with the restrictions of hardware development, such as the only limited subsequent adaptability of the increments (Engesser and Claus, 1993). A major difference between the development of mechatronic products and software results from the production time for prototypes, which is significantly longer for hardware and thus also for mechatronic products as a whole than for software (Schmidt et al., 2017a).

3.2 Agile approaches in product development

3.2.1 Kanban

The method Kanban to reduce waste by minimizing overproduction was originally developed in the course of the Toyota production system. The Japanese word Kanban originally means signal card. These signal cards were used during production to create an even flow during just-in-time production. This is done by the so-called pull effect. The material flow is created by a previous demand for parts, i.e. the drawing of a Kanban card. (Sugimori et al., 1977) This approach was later adopted by David J. Anderson to create a method for optimizing software development. (Anderson and Carmichael, 2016) The Kanban-Board helps to visualize the workflow and to communicate priorities to the team. In addition, Bottlenecks can be identified quickly and easily. Furthermore, the work in process is limited and the workflow is measured continuously in order to increase productivity. Thus, the focus of the development is placed on the development of value-adding elements for the customer in addition to the minimization of waste. (Matharu et al., 2015)

3.2.2 Scrum

The agile project management method Scrum has its origin in the field of software development and serves to support small project teams in the development process (Rising and Janoff, 2000). Scrum is based on the values transparency, inspection and adaptation, which are operationalized by a

framework of events, roles and artefacts (Schwaber and Sutherland, 2017). Products are developed by continuously dividing complex problems into manageable work packages and processing them. Products are developed incrementally. The development team carries out the actual development, the scrum master acts as method coach and the product owner communicates the stakeholder goals to the development team and is responsible for the targeted product development (Gloger, 2016). Development team, Scrum master and product owner form the Scrum team. In Sprint Planning, the work packages to be processed in the coming Sprint are defined and prioritized. These are converted in the fuel into a product increment that must be potentially shippable. The Daily Scrum takes place daily during the sprints and serves to discuss the current state of development and any problems in the development team. The increment is accepted in the Sprint Review by the product owner and the stakeholders. In the Retrospective Meeting the Scrumteam focuses on the optimization of the own development process. All developed results within a sprint are tested directly by the members of the development team. Through the iterative approach according to Scrum, the team is able to continuously validate all results, derive and prioritize new goals (by the product owner) and finally implement them. As a result, Scrum teams are characterized by a high reactivity to dynamically changing stakeholder requirements. (Schwaber and Sutherland, 2017)

3.2.3 SAFe

The Scaled Agile Framework for Lean Enterprises (SAFe) is a scalable and configurable framework. This framework helps companies deliver new products, services and solutions in the shortest possible time and with the best possible quality. SAFe combines approaches from the agile methods Scrum, Kanban and Extreme Programming with Lean-Agile principles and values. SAFe operates on four levels - Portfolio, Value Stream, Program and Team. At the team level, Scrum acts in a similar way to the project management method. Together with the program level, the team level forms an organizational structure called Agile Release Train (ART), in which agile teams, key stakeholders and other resources are dedicated to an important, ongoing solution mission. The portfolio level contains the principles, practices and roles required to initiate and manage a set of development value streams. It defines strategy and investment financing for value streams and their solutions. This approach provides guidance on roles, responsibilities, artifacts and activities to achieve better business results. SAFe enables agility to be applied in the enterprise environment and on a large scale. (Scaled Agile, 2017)

3.2.4 Design thinking

Design Thinking is a systematic approach to complex problems that has been developed from practical experience in order to develop user- and customer-oriented solutions. Thus Design Thinking represents a process and a mindset whose problem-solving potential enables a lively culture of innovation. (HPI School of Design Thinking, 2018). Design Thinking aims at innovations that combine three essential components. These are technological feasibility, economic viability and human desirability. During the iterative process, the human perspective serves as the starting point for the objective of developing new products or services that are marketable and technologically feasible. (Plattner et al., 2011)
The three proven success factors of the Design Thinking approach are the iterative process, a collaborative working and thinking culture of multidisciplinary teams and creative working environments. The design thinking process mentioned above consists of seven different phases, which are run through in iterative loops. These phases can be assigned to the overall goals of problem definition, idea generation and implementation. (Uebernickel et al., 2015) Various methods are used within the Design Thinking phases and great value is placed on visualization and prototyping. In particular, by integrating the problem solving method of Systems Thinking, which focuses on feedback to solve complex problems and increase agility. (Lewrick et al., 2017)

3.2.5 TAF-model

The TAF Agile Framework supports mechatronic system development as a methodical framework divided into the three areas desirebility, feasibility and viability. These are implemented as independent PDCA cycles. Starting with a product vision, products are developed incrementally per cycle up to a final prototype that represents the maximum intersection of the three cycles. The framework supports developers by providing different artifacts (e.g. user stories) in the continuous reduction of uncertainties. While the desirability cycle focuses on the validation of user stories, the feasibility cycle evaluates the technical feasibility of products and the viability cycle evaluates the

suitability of the respective concept for a functioning business model. Thus an iterative development of concepts takes place on functional as well as customer and market level. (Spreiter *et al*., 2018)

3.2.6 Lean start-up

Lean Startup is a business start-up approach where all processes, including all experiments, are kept as lean as possible and progress is measured in an environment of extreme uncertainty. Blank and Dorf (2012, p. XVII) define a startup as a "temporary organisation in search of a scalable, repeatable, profitable business model".

In Comparison to traditional strategic planning, the Lean Startup method involves conducting experiments from day one and developing real products that are tested on real customers instead of developing them internally. Therefore, finding the right business model is the result of experimentation, while finding the right product is the step of experimentation. It also provides unbiased feedback in a fraction of time compared to traditional market analysis. (Ries, 2011)

3.3 Existing comparisons of agile approaches

Agile approaches have already been subjected to numerous comparisons in the literature (especially with classical, plan-driven approaches) (Barlow *et al*., 2011). An established model for comparing agile approaches is the 4-DAT evaluation framework, which compares the approaches with regard to the categories *Method Scope*, *Features* (*Agility Characterization*), *Agile values* and *Software Process Characterization*. However, since the focus of 4-DAT is clearly on software development, an agile approach to mechatronic systems development lacks the implications that physical products and large organizations have on the level of agility of each approach. For example, approaches that are flexible and lean and increase development speed are classified as more agile than approaches that do not have these characteristics (Qumer and Henderson-Sellers, 2008). Often, the agile approach of extreme programming (Lassenius *et al*., 2015) is used as a basis for comparison. In particular, the comparison focuses on the handling of software requirements, the minimization of complexity, responsiveness to changes in the development context and the integration of verification techniques. (Fernandes and Almeida, 2010) In addition, agile approaches are often discussed with regard to the implementation of the principles from the agile manifesto. (Fowler and Highsmith, 2001) For the context of mechatronic system development, however, these comparisons are only conditionally useful, since the comparison criteria always originate from the context of software development.

4 RESULTS

4.1 Cluster-analysis of the term agility

In order to analyse the term agility, a cluster analysis was executed with 3712 papers, which were published on the *International Conference on Engineering Design* in the years from 2002 to 2017.
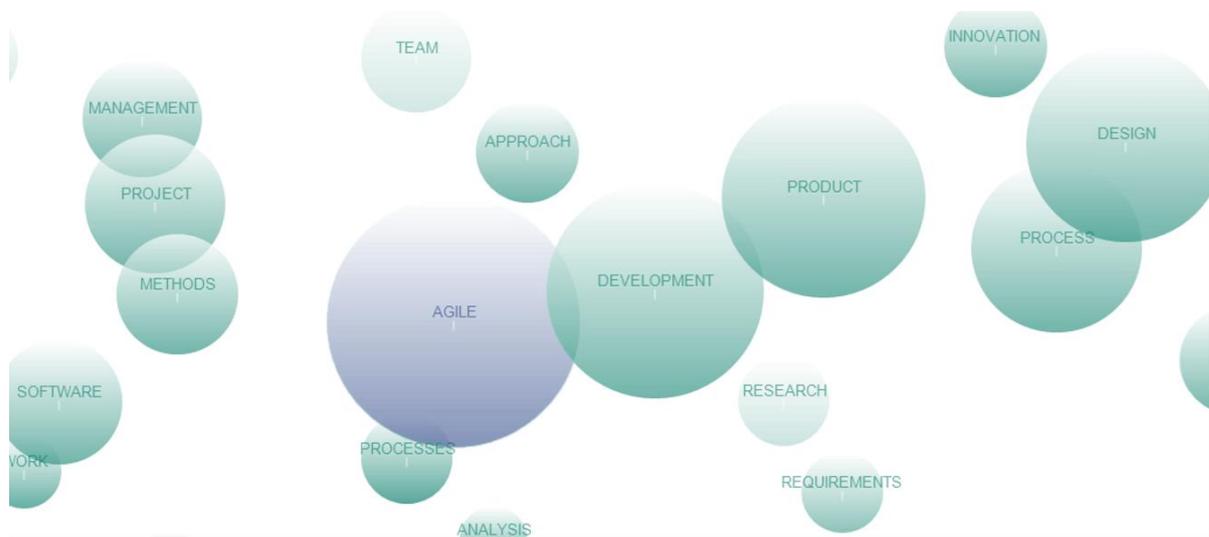

Figure 1. Immediate surrounding of the term agile in a cluster analysis

For the purpose of evaluating the papers, the QDA Miner was used. The QDA Miner is a software package for qualitative data analyses. It is easy to use and represents a tool for coding, recalling, analysing and commenting on small as well as big collections of documents and images. (Provalis Research, QDA Miner 5) In the first step, all passages which contain the terms *agile* and/or *agility* were filtered from the papers. With the help of WordStat (Provalis Research, WordStat 6) – a software for content analysis and text mining – all relevant paragraphs can be evaluated. In order to determine the terms, which are used in context with the term agility, a bubble chart – which is presented in Figure 1 – was created. The chart shows all terms, which are found the most often in context with the term *agile*. Furthermore, a table to determine the most frequent phrases, which occur in the selected paragraphs, was created. Another table was compiled in order to list the phrases, which are often found in the selected paragraphs. Phrases which are not relevant for the evaluation – like *International Conference*, *Case Study* or *International Journal* for example – were removed from the list. Figure 2 presents the results of this evaluation in which the most frequent relevant phrases are compared to the frequency of their occurrence.
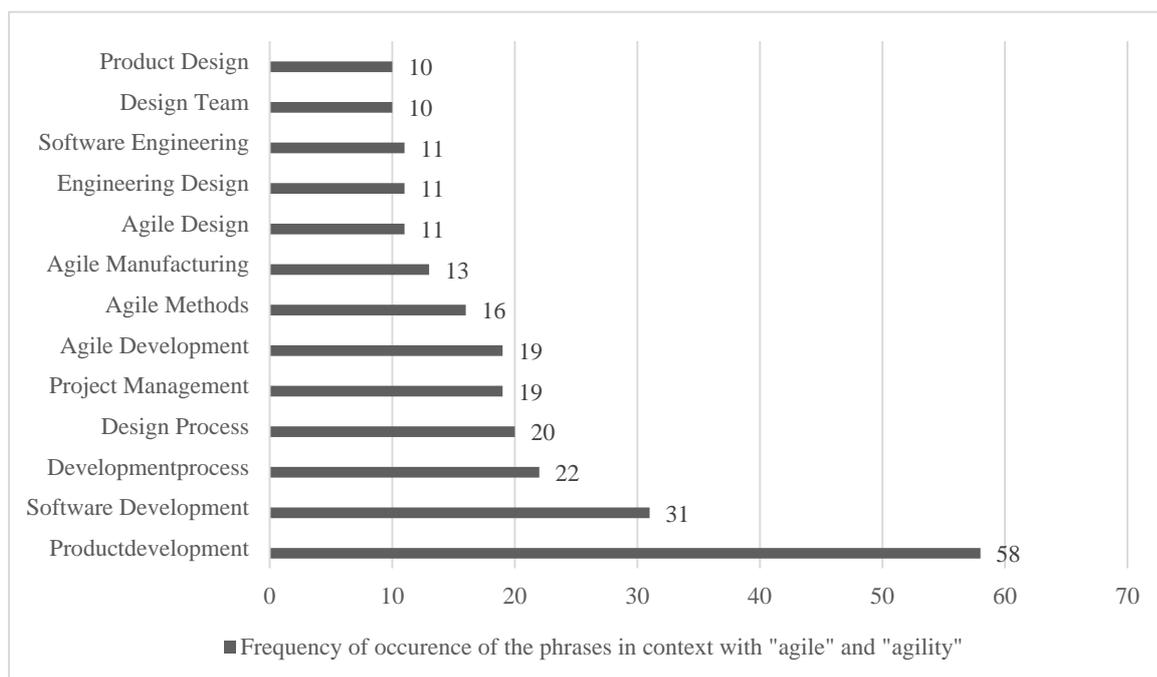

Figure 2. Frequency of occurrence of relevant phrases in context with agile and agility

The term agility is used the most often in context with the development and the design of products and software. It is also often found in context with processes, project management and manufacturing while *product design* and *design team* are the least used with this term. In general, processes of development and design are strongly connected with the term agility. It becomes also clear that agility plays a big role in project management as well. In contrast to this, the term is used less in context with manufacturing.

4.2 Comparison of the agile approaches in context of development of mechatronic systems

The approaches were compared with regard to different factors from the literature (Gericke *et al.*, 2013) which, if addressed by certain mechanisms, support the development of mechatronic systems according to the situation and requirements. Initially in an expert workshop, 19 factors were identified as most relevant for mechatronic system development. These come from the clusters *understanding of roles*, *adaptability*, *customer integration*, *use of technical and process-related knowledge*, *system architecture*, *controlling* and *method provision* (see Figure 3, 1st column). In the analysis of the approaches, these were evaluated with regard to their integrated mechanisms for satisfying the respective factor. It was found that no approach serves all factors simultaneously. This is due to the different issues for which the respective approaches have been designed. Each approach is excellent for supporting development teams in dealing with the issue at hand.

Factor	Kanban	Scrum	SAFe	Design-Thinking	TAF-Modell	Lean Start-Up
Assignment of roles	There are no required roles, two functions may be helpful: Service Request Manager, Service Delivery Manager	Scrummaster, Product-Owner, Development Team	Team-Level: Scrummaster, Product-Owner, Development Team Program-Level: System and Solution Architect/Engineering, Product and Solution Management, Release Train Engineer and Solution Train Engineer, Business Owners Portfolio-Level: Epic Owner, Enterprise Architect	Design Thinking Coach, Development Team	Not adressed	"Product-Owner", Development Team (Start-Up)
Project team composition	No mechanism known	Crossfunctional team with all necessary competences	Team-Level: Crossfunctional team with all necessary competences Program-Level: Crossfunctional team with all necessary competences Portfolio-Level: Functional team with specific competences	Multidiciplinary team with all necessary competences	Not adressed	Crossfunctional team with all necessary competences
Controlling project progress	Kanban board visulizes work in progress and project flow	Burndown-Chart, Daily Scrum	Team-Level: Burndown-Chart, Daily Scrum; objective measures, including average lead time, WIP and throughput Program-Level: Continuous Delivery Pipeline Portfolio-Level: Milestones	No Mechanism known	Not adressed	MVP, Hypotheses, interactive product launch
Mechanisms for continuous improvement of the process	Feedback Loops/ 7 cadences	Retrospective Meeting, Daily Scrum	Team-Level: Retrospective Meeting, Daily Scrum Program-Level: No Mechanism known Portfolio-Level: Possible on Milestones	Integration of Feedback (but no fixed time of feedback)	Integrated via PDCA-cycle	Short development-cycles , early customer feedback
Mechanisms of requirements management	Pool of ideas	Product-Backlog	Team-Level: Team Backlog Program-Level: Program and Solution Backlog Portfolio-Level: Strategic Themes, Portfolio Canvas and Backlog, Guardrails	No Mechanism known	Methods in feasibility and desirability layers are recommended	Short development-cycles , early and iterative customer feedback
Consideration of the system architecture	No Mechanism known	Scrum at Scale, Scrum of Scrum	Team-Level: No Mechanism known Program-Level: No Mechanism known Portfolio-Level: No Mechanism known BUT in Full SAFe - Large Solution: Integration of MBSE	No Mechanism known	No Mechanism known	MVP, (No Mechanism known)
Structuring the system architecture	No Mechanism known	No Mechanism known	Team-Level: No Mechanism known Program-Level: No Mechanism known Portfolio-Level: No Mechanism known BUT in Full SAFe - Large Solution: Integration of MBSE	No Mechanism known	Method on feasibility layer is recommendet	No Mechanism known
Flexible adjustment of the phase or sprint goal	No Phases or Sprints mentioned.	Sprint Goal is defined in Sprint planning and continuously controlled	Team-Level: Sprint Goal is defined in Sprint planning and continuously controlled Program-Level: Iterative process Portfolio-Level: sequential process	Phase Goal is defined and controlled by the DT-Coach	Via adaptable PDCE-cycles	Early and iterative customer feedback leads to the goal of the next dev.-cycle
Continuous integration of developed increments into the overall system	Incremental Development	Incremental Development	Team-Level: Incremental Development Program-Level: Incremental Development Portfolio-Level: No mechanism known	No Mechanism known, based on experiences of Development Team	Intersection of the three views feasibility, viability and desirability	Incremental Development
Parallelization of activities	Possible to visualize on Kanban-Board	Possible by using Burndown-Chart	Operating on different levels	Possible through the iterativ process	Via PDCA-Cycles	Parallel experiments possible
Addressing different views (management/controlling, method selection, method implementation)	No Mechanism known	No Mechanism known	Integration of different roles and perspectives through Program- and Portfolio-Level	Considered by multidisciplinary Teams as well as conducting interviews	Feasibility, viability and desirability	Considered by multidisciplinary Teams as well as conducting interviews
Continuous integration of the customer perspective	Customer Value as a Kanban-Value	Epics and User-Stories, Product Owner = Customers Voice	Epics, User-Stories and Customer	360° Research (Intervievs, Observations, market research,..) Persona Method, User-Stroies, Prototyp validation	Via layer desirability	Continuosly, early and iterative customer feedback
Differentiation between customer and user view	Not operated actively	Not operated actively	Not operated actively	Not operated actively	Not operated actively	Not operated actively
Provision of mechanisms for the integration of existing process knowledge in the company	No mechanism known.	Sprint 0: Aims at finding the delta, No Mechanism known, based on experiences of Development Team	No Mechanism known, based on experiences of employees	Scoping, 360° Research,Synthesis: based on experiences of Development Team	No Mechanism known	MVP: requested delta, No Mechanism known, based on experiences of Development Team
Provision of mechanisms for the integration of knowledge into products	No mechanism known.	Sprint 0: Aims at finding the delta, No Mechanism known, based on experiences of Development Team	There are several methods integrated in each level	Scoping, 360° Research,Synthesis: based on experiences of Development Team	Some Methods are recommended	MVP: requested delta, No Mechanism known, based on experiences of Development Team
Provision of methods for the analysis and generation of new objects	No mechanism known.	No Mechanism known, based on experiences of Development Team	No Mechanism known, based on experiences of Development Team	Variety of methods during the phases	Few Methods are recommended	No Mechanism known, based on experiences of Development Team
Mechanisms for early and continuous validation of development results	Cadences regarding customer deliveries	Sprint-Review	Team-Level: Sprint-Review Program-Level: No mechanism known Portfolio-Level: No mechanism known	Early Prototyping and Validation Phases (iteratively)	Validation from the three views feasibility, viability and desirability	Experiments from day one with direct customer feedback to product-launch (prototyp)
Scalability over project duration	No restriction of the method by project duration	Fixed timed sprints (approx. 4 weeks), number flexibly adaptable to the project	Fixed timed sprints (approx. 4 weeks), number flexibly adaptable to the project	Fully scalable	Possible	Very short development-cycles, repetition scalable
Scalability on different organizational project structures	No problems by skaling Kanban	Large Scaled Scrum	Essential, Large Solution, Portfolio and Full SAFe	No	No Mechanism known	No

Figure 3. Comparison of agile approaches by different factors

The identification of the relevant factors showed that a large number of factors were independent of the product developed (software or mechatronics). However, since software is a logical system and hardware is a physical system, there are 6 factors that result in particular from this distinction (Youn and Yi, 2014) (see Figure 3, factors highlighted in grey in the first column).

In Figure 3, each factor was assigned to the respective mechanisms of the individual approaches, insofar as these could be identified from the literature. In addition, a colour scheme was used to get a quick overview of potentials and limits. Meaning:

- green lines, that the respective approach provides adequate mechanisms for the satisfaction of the associated factor
- yellow, that isolated mechanisms exist which, however, were not explicitly generated to satisfy the factor or that certain aspects are addressed exclusively by experiences of developers
- dark red, that no mechanism for the satisfaction of the respective factor could be identified

All within this publication analyzed agile approaches were designed to satisfy a concrete purpose. For example, Kanban aims to minimize power dissipation in development; Design Thinking aims to promote creative potential in development and thus support customer-oriented solution finding. The TAF model is the only approach designed for mechatronic system development. All other approaches were primarily designed for software development. The purpose-bound characteristic of the approaches becomes clear with the analysis of the identified factors. Many of the analysed approaches include different mechanisms for customer integration or continuous validation of development results. They are also well suited for scaling across different project durations or organizational levels. The implementation of incremental development results and the continuous integration of these into the overall system is also accelerated by the predominant number of approaches.

However, there is potential, especially in the identification and integration of existing process knowledge, which could be supported by mechanisms such as the creation and continuous further development of reference process models. In addition, with the exception of SAFe, the approaches do not sufficiently integrate the system architecture to support mechatronic system development adequately. The same applies to the conscious integration of existing product knowledge into current developments. Here the integration of the mindset that products are developed in generations and existing solutions from other contexts can be integrated into the development of the respective product is suitable. In addition, it is also conceivable that methods could be applied over the development of different generations.

The consideration of development projects as integrated was not considered by the factors. However, in industrial practice, the validation and production system is continuously co-developed alongside the product. For example, the design and construction of certain application-specific test benches sometimes takes more than a year. These aspects must also be considered when agile approaches have to be applied in mechatronic system development. The goal for development teams must not be to develop agile until they have identified a relevant customer need or secured a certain functional principle. Rather, agile approaches must also integrate the detailed design, such as the dimensioning of components or the design of the necessary surface quality in a shaft-hub connection, into agile development. Here, too, the continuous inclusion and conscious application of technical knowledge can support.

5 DISCUSSION AND OUTLOOK

In this article, different approaches to support agile development projects were examined and compared with regard to their suitability for the context of mechatronic system development. Based on an analysis aimed at identifying research priorities in the field of agility, factors were derived that would provide promising support for the development of mechatronic systems. The agile approaches were evaluated with regard to these factors and potentials and limitations of the approaches were identified. It was found that the approaches are well suited for their respective purposes. However, they lack the conscious integration of technical or process-related knowledge. In addition, the integrated consideration of product development and the simultaneous development of associated validation and production systems is currently not addressed by existing agile approaches.

Since the approaches in this paper were compared with regard to factors from the literature, the aim of future research contributions must be to validate and weight the factors with representatives from practice. In addition, this article provides the basis for a deeper examination of the interrelationships between certain factors. Thus, approaches must be derived which, on the one hand, take into account

the respective complexity of the existing development context and, on the other hand, integrate mechanisms to continuously integrate process and product knowledge into the process of product development. Harmonisation with the development of validation and production systems should also be sought. The findings in this article will be used in future work to shape the approach of ASD - Agile Systems Design (Albers *et al*., 2019) with regard to targeted application in the agile development of mechatronic systems. The basic principles identified in ASD are operationalised by appropriate practices with regard to the fulfilment of the factors identified in this paper. In addition, follow-up work should deal with cross-domain requirements management and cross-domain interface conflicts in order to support consistency in agile development.

REFERENCES

Albers, A., Heimicke, J., Spadinger, M., Reiss, N., Breitschuh, J., Richter, T., Bursac, N. and Marthaler, F. (2019), "Eine Systematik zur situationsadäquaten Mechatroniksystementwicklung durch ASD – Agile Systems Design", in *KIT Scientific Working Papers*, Vol. 113, KIT, Karlsruhe. https://doi.org/10.5445/IR/1000091847

Anderson, D.J. and Carmichael, A. (2016), *Essential Kanban condensed, Essential Kanban*, Vol. 2, First edition, Lean Kanban University Press, Seattle, Washington.

Barlow, J., Giboney, J., Keith, M.J., Wilson, D., Schuetzler, R., Lowry, P.B. and Vance, A. (2011), "Overview and Guidance on Agile Development in Large Organizations", *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1909431

Blank, S. and Dorf, B. (2012), *The Startup Owner's Manual: The Step-by-Step Guide for Building a Great Company*. K&S Ranch.

Broy, M. (Ed.) (2010), *Cyber-Physical Systems: Innovation durch Software-Intensive Einebette Systeme*, Springer, München. https://doi.org/10.1007/978-3-642-14901-6

Cooper, R.G. (2015), "What's Next? After Stage-Gate", *Research-Technology Management*, Vol. 57 No. 1, pp. 20–31. https://doi.org/10.5437/08956308X5606963

Czichos, H. (2015), *Mechatronik: Grundlagen und Anwendungen technischer Systeme*, Springer Vieweg, Berlin. https://doi.org/10.1007/978-3-658-09950-3

Engesser, H. and Claus, V. (1993), *Duden Informatik ein Sachlexikon für Studium und Praxis*, Dudenverlag, Mannheim.

Fernandes, J.M. and Almeida, M. (2010), "Classification and Comparison of Agile Methods", in Abreu, F.B.e. (Ed.), *Seventh International Conference on the Quality of Information and Communications Technology (QUATIC)*, IEEE, Piscataway, NJ, pp. 391–396. https://doi.org/10.1109/QUATIC.2010.71

Fowler, M. and Highsmith, J. (2001), "The Agile Manifesto", *Softw. Dev.*Vol. 9, pp. 28–35.

Gericke, K., Meißner, M. and Paetzold, K. (2013), "Understanding the context of product development", *Proceedings of the 19th International Conference on Engineering Design (ICED13) Design For Harmonies*, Vol. 75 No. 3.

Gloger, B. (2016), "Die Rollen – klare Verantwortlichkeiten", in Gloger, B. (Ed.), *Scrum: Produkte zuverlässig und schnell entwickeln*, Carl Hanser Verlag, München, No. 5, pp. 59–107. https://doi.org/10.3139/9783446448360.002

Hasso-Plattner-Institut (HPI) (2018), *Design Thinking. Mindset*. Retrieved on 20.07.2018 at https://hpi.de/school-of-design-thinking/design-thinking/mindset.html.

ISO/IEC/IEEE 24765:2010: Systems and software engineering - Vocabulary (2010).

Laamanen, T., Lamberg, J. and Vaara, E. (2016), "Explanations of Success and Failure in Management Learning: What Can We Learn From Nokia's Rise and Fall?", *Academy of Management Learning & Education*, Vol. 15 No. 1, pp. 2–25. https://doi.org/10.5465/amle.2013.0177

Lassenius, C., Dingsøyr, T. and Paasivaara, M. (Eds.) (2015), *Agile Processes in Software Engineering and Extreme Programming*, Vol. 212, Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-18612-2

Lewrick, M., Link, P. and Leifer, L. (2017), *Das Design Thinking Playbook*. Vahlen, München.

Matharu, G.S., Mishra, A., Singh, H. and Upadhyay, P. (2015), "Empirical Study of Agile Software Development Methodologies", *ACM SIGSOFT Software Engineering Notes*, Vol. 40 No. 1, pp. 1–6. https://doi.org/10.1145/2693208.2693233

Plattner, H., Meinel, C., Leifer, L. (2011), *Design Thinking. Understand-Improve-Apply*. Springer Verlag, Berlin/Heidelberg.

Provalis Research, User's Guide QDA Miner 5

Provalis Research, User's Guide WordStat 6 - Content Analysis Module for QDA Miner and SimStat

Qumer, A. and Henderson-Sellers, B. (2008), "An evaluation of the degree of agility in six agile methods and its applicability for method engineering", *Information and Software Technology*, Vol. 50 No. 4, pp. 280–295. https://doi.org/10.1016/j.infsof.2007.02.002

Rebentisch, E., Conforto, E.C., Schuh, G., Riesener, M., Kantelberg, J., Amaral, D.C. and Januszek, S. (2018), "AGILITY FACTORS AND THEIR IMPACT ON PRODUCT DEVELOPMENT PERFORMANCE", in *International Design Conference – Design 2018*, pp. 893–904. https://doi.org/10.21278/idc.2018.0236

Ries, E. (2011), *The Lean Startup-How Constant Innovation Creates Radically Successful Businesses*. Crown Business, United States of America.

Rising, L. and Janoff, N.S. (2000), "The Scrum software development process for small teams", *IEEE Software*, Vol. 17 No. 4, pp. 26–32. https://doi.org/10.1109/52.854065

Royce, W.W. (1970), "Managing the Development of Large Software Systems", *Proceedings of IEEE WESCON*, pp. 1–9.

Scaled Agile (2017), "SAFe 4.5 Introduction, Overview of the Scaled Agile Framework for Lean Enterprise", *White Paper*, p. 1.

Schmidt, T.S., Chahin, A., Kößler, J. and Paetzold, K. (2017a), "Agile development and the constraints of physicality: A network theory-based cause-and-effect analysis", *21ST INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED17*.

Schmidt, T.S., Weiss, S. and Paetzold, K. (2017b), "Agile Development of Physical Products. An Empirical Study about Motivations, Potentials and Applicability", *University of the German Federal Armed Forces*.

Schmidt, T.S., Weiss, S. and Paetzold, K. (2018), "EXPECTED VS. REAL EFFECTS OF AGILE DEVELOPMENT OF PHYSICAL PRODUCTS. APPORTIONING THE HYPE", in *INTERNATIONAL DESIGN CONFERENCE - DESIGN 2018*, pp. 2121–2132. https://doi.org/10.21278/idc.2018.0198

Schöner, H.-P. (Ed.) (2006), *Handbuch der Mess- und Automatisierungstechnik im Automobil: Mechatronik*, Springer-Verlag, Berlin, Heidelberg.

Schuh, G., Dölle, C., Kantelberg, J. and Menges, A. (2018), "Identification of Agile Mechanisms of Action as Basis for Agile Product Development", *Procedia CIRP*, Vol. 70, pp. 19–24. https://doi.org/10.1016/j.procir.2018.02.007

Schumpeter, J.A. (1912), *Theorie der wirtschaftlichen Entwicklung.*, 1st ed., Verlag von Duncker & Humblot, Leipzig.

Schwaber, K. and Sutherland, J. (2017), "The Scrum Guide".

Smith, P.G. (2007), *Flexible Product Development: Building Agility for Changing Markets by*, John Wiley & Sons.

Spreiter, L., Böhmer, A.I. and Lindemann, U. (2018), "EVALUATION OF TAF AGILE FRAMEWORK BASED ON THE DEVELOPMENT OF AN INNOVATIVE EMERGENCY WEARABLE FOR SENIORS", in *DS92: Proceedings of the DESIGN 2018 15th International Design Conference*, pp. 1345–1356. https://doi.org/10.21278/idc.2018.0252

Sugimori, Y., Kusunoki, K., Cho, F. and Uchikawa, S. (1977), "Toyota production system and Kanban system Materialization of just-in-time and respect-for-human system", *International Journal of Production Research*, Vol. 15 No. 6, pp. 553–564. https://doi.org/10.1080/00207547708943149

Tuominen, M., Rajala, A. and Möller, K. (2004), "How does adaptability drive firm innovativeness?", *Journal of Business Research*, Vol. 57 No. 5, pp. 495–506. https://doi.org/10.1016/S0148-2963(02)00316-8

Uebernickel, F., Brenner, W., Pukall, B., Naef, T. und Schindelholzer, B. (2015), *Design Thinking. Das Handbuch*. Frankfurt Allgemeine Buch, Frankfurt am Main.

VDA - Verband der Automobilindustrie (2007), *Qualitätsmanagement in der Automobilindustrie: Automotive SPICE Prozessaassesment*, 1st ed.

Wynn, D.C. and Eckert, C.M. (2017), "Perspectives on iteration in design and development", *Research in Engineering Design*, Vol. 28 No. 2, pp. 153–184. https://doi.org/10.1007/s00163-016-0226-3

Youn, W. and Yi, B. (2014), "Software and hardware certification of safety-critical avionic systems: A comparison study", *Computer Standards & Interfaces*, Vol. 36 No. 6, pp. 889–898. https://doi.org/10.1016/j.csi.2014.02.005