AGILE MEETS PLAN-DRIVEN – HYBRID APPROACHES IN PRODUCT DEVELOPMENT: A SYSTEMATIC LITERATURE REVIEW

J. Heimicke ✉, R. Chen and A. Albers

Karlsruhe Institute of Technology, Germany

✉ jonas.heimicke@kit.edu

Abstract

More than 15 years after the publication of the agile manifesto of software development, agile development approaches have also reached the processes of physical product development. Because of the boundary conditions and requirements here, which differ strongly from those of pure software development, these approaches often reach their limits. However, research and practice have quickly recognized that hybrid approaches integrate the strengths of agile and plan-driven development. This paper presents 25 hybrid development approaches that have been identified in a Systematic Literature Review.

Keywords: product development, design process, engineering design, agile product development, hybrid development approaches

1. Introduction

The integration of agile and traditional development is gaining more and more attention for a variety of reasons. In the development of mechatronic products, the proportion of the three major engineering sciences involved - mechanical engineering, electrical engineering, and computer science - are roughly equal. As an interdisciplinary product development, to better meet customer requirements, the disciplines in various fields should be further integrated. (Neumann, 2012) In order to meet the requirements of interdisciplinarity in development, companies used to adopt some traditional development models, such as V-model. However, to thoroughly consider changing markets, variable customer requirements, the Internet of Things, and many other challenges, companies use new approaches - agile approaches - in development. (Schmidt et al., 2019) Although many companies are still implementing agile methods, (Goevert et al., 2019) some companies have noticed that pure agile methods in mechatronics system development have reached their limits (for example, the following shortcomings: lack of necessary documents; measuring the progress of the entire project is difficult). (Heimicke et al., 2019) For this reason, they often combine agile and traditional approaches. For instance, Cooper and Sommer (2018) found that today's gated process is too rigid, and an overly strict development model makes the development process less active and slow to respond. But markets and technologies are getting faster and faster, and there are global factors. Companies must release products and bring them to the market in a short period. Product development is more challenging than before. They built a hybrid model - Agile-Stage-Gate - that has the advantages of both the structure and control of the stage-and-gate system and the speed and productivity of the agile method. (Cooper and Sommer, 2018) Furthermore, Hybrid Development Approaches (HDA) aim to choose a process

model to fit best a product development project, which is also possible to do this on a higher level of detail. (Schuh et al., 2018) As mentioned earlier, there are advantages and disadvantages to agile development and traditional methods. Some researchers and some companies have realized that the hybrid method created by combining the two is a feasible way. The hybrid development approach has great prospects and potential in the mechatronics industry. However, as the number of hybrid approaches continues to rise, there is no comprehensive interpretation of the various development approaches, such as composition, development framework, and practical examples. This is the main reason for our literature review to further develop hybrid methods with specific applications in different environments, and we need to better understand its composition, development framework, and practical examples. The purpose of this article is to lay the foundation to further hybrid development methods through a literature review that describes the main attributes of hybrid development and studies the advantages and challenges of these methods. In this way, we provide an overview of the current state of research in the field of hybrid development approaches. The understanding created in this way enables the further development of the situation- and demand-oriented use of suitable development approaches in later research work.

2. Plan-driven and agile development

New product development is the source of competitive advantage for enterprises and the basis for the continuous growth of corporate profits. (Schumpeter, 1912) In this fast-changing era, industrial companies, from the assumption of the user's needs to the official production of the product to the market, which goes through many stages, involving a wide range of technical fields and many steps, so it must work according to specified procedures. To support developers in the product development process, a distinction is made between traditional or plan-driven procedures and agile procedures. Plan-driven methods include, besides others, the Waterfall model or the V model. Their characteristics are distinguished based on different development stages. The waterfall model is one of the most commonly used development methods. It is a linear structure, which means that only the previous phase be ultimately ended, and the next phase begins. It is characterized by simplicity and ease of management. Users can only see development results when they wait until the end of the process, which increases development risk through undetected errors that were made early in the process and have significant late effects. Traditional plan-driven models can be better applied to large projects. Strategies, documents, and processes can contribute to a better connection and coordination among large organizations. (Boehm and Turner, 2004) At the same time, agile development is a lightweight development that uses short-term iterations to engage users in projects actively, validate and prioritize requirements. Real agile development must be iterative, incremental, self-organizing, and emergent. This means that the project can take several cycles to complete, and each cycle delivers semi-finished products until it is completed. The team manages work most competently, and the process, principle, and work structure are determined during the project, rather than being pre-set at the outset. (Boehm and Turner, 2005) In mechatronic system development, agility is particularly apparent as the ability of a development team to adapt the phases of synthesis and analysis reactively and purposefully to changes in the development context. (Albers et al., 2019a) A variety of factors can distinguish agile development and plan-driven development. Agile development is designed for small development teams, close communication within the team, collaboration between developers and customers, continuous delivery of available samples in a short period, and optimization of products based on customer feedback. The plan-driven development relies on clear steps and terms. Its core lies in the definition and management of the process. The advantage is the repeatability brought by the standardized steps and the stability in the development process. It is suitable for large-scale team development projects based on customer requirements. (Boehm and Turner, 2004) Common Approaches of Agile Development are Scrum, eXtreme Programming and Kanban. Scrum is a framework for developing and maintaining complex products. It is an incremental, iterative development process. In this framework, the entire development process consists of several short iteration cycles. A short iteration cycle is called a Sprint, and the recommended length of each Sprint is 2 to 4 weeks (1-week Sprint can be used for Internet product development). In Scrum, the product backlog is used to manage the requirements of

the product. The product backlog is a list of requirements sorted by business value. The manifestation of the list entry is usually a user story. Extreme Programming focuses on specific engineering practices. XP is designed to enable developers to respond confidently to customer needs through a reasonable mix of engineering practices. Emphasize the feedback loop mechanism, the feedback loop between the customer and the R & D team, the feedback loop for testing and development, and the feedback loop between specific code implementation and unit testing. Kanban comes from *visual signals* or *cards* in Japanese. Toyota uses Kanban to control the production process. During the entire production process, these cards will be accompanied by separate materials, and the cards will send corresponding signals at different stages of production, indicating that this work step has been completed. However, the number of cards is limited, which can avoid production bottlenecks. If all cards are in use, new work cannot begin until the cards have completed the entire production process and are available again. (Saleh et al., 2019)

3. Research approach

Hybrid approaches in product development combine the strengths of agile and plan-driven approaches. However, there is currently no overview and analysis of existing hybrid approaches. In order to create a basis for further research on hybrid approaches in product development, the following research questions are answered in this paper:

RQ1: Which hybrid approaches to product development exist in the literature? What are their basic principles?

RQ2: What are the benefits of adopting Hybrid Development Approaches for the development process?

The method used in this research is the Systematic Literature Review (SLR). It was conducted in accordance with the guidelines provided by Kitchenham and Charters (2007). This guide covers the main steps of the literature review: identifying goals, defining methods, document collection, collating literature, researching features, and analysing results. (Kitchenham and Charters, 2007)

3.1. Data sources

In this paper, we searched in the following digital libraries and databases: Scopus und Web of Science (WoS). The reason why these two databases are selected is the following three points. According to Bauschmann and Ahnert (2017), WoS has some advantages, for example, deep development, improved regional coverage, self-citation can be excluded. Scopus is even more comprehensive than WoS. (Bauschmann and Ahnert, 2017) In order to get more inclusive data, we chose to search in both. Based on our research questions, we used the following string in both libraries.
(hybrid OR tailored) AND (hardware OR mechatronic OR software AND planning OR development OR process) AND (agile OR waterfall OR plan-driven).*

3.2. Inclusion and exclusion criteria

We aim at identifying papers that propose a method combining agile with traditional development methods, or the integration of an agile approach in the plan-driven development process. Besides, only English and peer-reviewed papers are included, and only one of several versions is retained. The following type of publications are excluded: papers proposing agile methods with a focus on Irrelevant content, articles offering approaches for the hybrid method with only Case Studies, papers proposing strategies for a hybrid method without addressing product development (e.g., hybrid material).

3.3. Data extraction

The search string obtains from SCOPUS and WOS, provides us with 1,628 articles. In Figure 1. we illustrate the selection process. The first step is to filter out the chapters in the book or book from the results obtained, and we get 1478 articles. The second step is to filter out the duplicates, and now there are 1445 contributions left. The third step is done by reading the title and abstract. Chose whether to continue reading the full documentation based on the exclusion criteria.
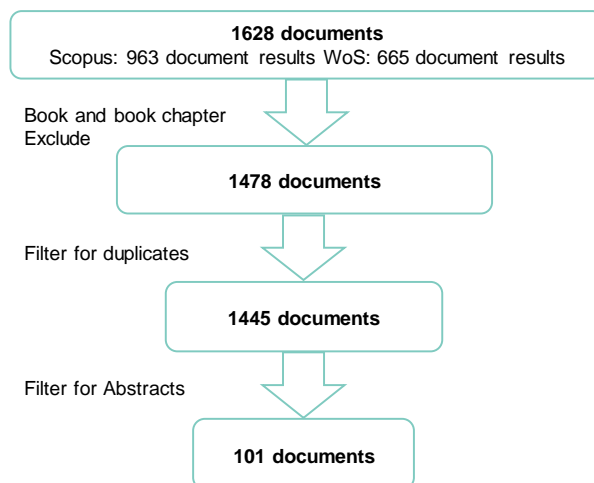
Figure 1. Selection process of the conducted Systematic Literature Review

4. Results

Based on the contents of the title and abstract, 101 contributions remained. We have analysed these and identified 25 significantly divergent hybrid approaches which are presented in Table 1. In addition to the core idea (column 3) of the respective approach, the second column shows the elements used to form the new approach. The original source is shown in the fourth column. Table 1 shows the central result of the research work. In addition, we show an excerpt from our analysis of hybrid approaches with regard to their benefits. Here is a brief introduction to the following several important methods of agile development.

Distribution of Papers: When investigating the result set, it became obvious that, while the Hybrid Development Approaches were first discussed in 2007 (User-centered agile), the topic gained momentum in 2014 and, since then, a continually high number of relevant articles have been published every year. This reflects that, while the idea of integrating traditional and agile development has been around for some time, integration problems are still unconcluded and investigation is continuing.

Table 1. Overview of Hybrid Development Approaches in this SLR

Approaches and Main Ideas	Elements	Source	No.
Agent Agile Game Development methodology: The suggested methodology AAGDM combines agile methodology that meets the dynamic requirements of the customer with Multi Agent Software Engineering (MaSE). The MaSE used in the Sprint phase is at the core of the AAGDM.	Daily Scrum Sprint MaSE	(Al-Azawi et al., 2014)	A1
Agile & JAD: In this paper, they are providing a hybrid approach for requirement engineering in the agile with the help of Joint Application Development (JAD) and the prioritization of the requirements in the agile is helped by the viewpoint.	Daily Scrum JAD	(Kumar et al., 2014)	A2
Agile Model Driven can optimize software development, the most important is the higher yields and better-quality code. Second, more communication with customers during the development process. Its life development cycle is divided into three consecutive iterations.	Model Driven Sprints Daily Scrum	(Zhang and Patel, 2010)	A3
Agile Product Line Engineering: AgiFPL is designed to increase agility within the PLE and effectively meet any emerging business expectations. On the one hand, for the Domain Engineering phase, AgiFPL implements an iterative process that uses the i* method with Scrumban. On the other hand, for the Application Engineering phase, AgiFPL also implements an iterative process that uses the Scrum method in conjunction with i*. Each process of AgiFPL is based on iterative and incremental development.	Daily Scrum Product Line Engineering	(Haidar et al., 2017)	A4
Agile-Stage-Gate combines classic stage-gate and agile. The stage-gate process structure (stage and gate) and short-cycle iteration of agile development are designed to improve communication, quick feedback on customer needs, and higher productivity. And make development activities no longer isolated.	Stage-Gate Daily Scrum Sprints	(Cooper and Sommer, 2018)	A5

Agile-Waterfall: Water defines the upfront work. Teams use scrum to develop software in the middle of the process. Fall means establishing gates to limit software release frequency.	Waterfall Daily Scrum	(West, 2011)	**A6**
AI techniques & Agile: The model is made up of three layers: RUP layer, integrated Scrum layer and CBR (Case Based Reasoning) layer. Common CBR is an AI technique that emphasizes the memorization of prior cases. In the first place, the main RUP layer is founded on the overall values of a unified process (UP), which include cycle tracking, stakeholder preferences, team collaboration, repeatable validation, and stress on quality and fostering the level of detail. In the second place, the Scrum layer is merged with the fundamental RNP steps, namely: inception, elaboration, construction and transition. CBR is linked to the Scrum activities. The vision in shape of a use case diagram of the Unified Modeling Language (UML) is stored in CBR in the form of a case. Likewise, the deliverables and the solution are stored in CBR at the end, which can be used again in the future when the same shape of the vision occurs.	Daily Scrum AI technique	(Mukhtar et al., 2013)	**A7**
Competitor Driven Development & Extreme programming: Reverse engineering of competitor products was applied. List down all the better and attractive features present in it to incorporate them in our product. Test-First practice of XP ensures that the code is properly tested for functioning correctly.	XP FDRD	(Doshi and Patil, 2016)	**A8**
Design thinking & Extreme programming: The several Design thinking practices are adapted into two XP phases called the investigation and plan phase. - DT integration in XP exploration phase (Empathy, Define, User persona, DT-user stories); DT integration in XP planning phase (Automated prototyping, Prototype usability evaluation, Usability, User testing)	XP Design thinking	(Sohaib et al., 2019)	**A9**
Enterprise Architecture Management & Agile: This theory aims to use agile methods in EA management. Roles, duration, activities, and expected results are borrowed directly from Scrum. Map Scrum concepts (sprints, requirements, user stories) into EA management.	Daily Scrum EAM	(Hanschke et al., 2015)	**A10**
Global Software development & Agile: They suggest incremental integration, short iterations, regular short status meetings, various communication modes, customer representatives in offshore teams. Basic agile principles of communication mechanism, frequent deliveries are suggested to improve customer trust and relationship and bridge the gaps in coordination and also overcome cultural differences.	Global Software development	(Ivček and Galinac Grbac, 2008)	**A11**
Human Centered Design & Agile: User exploratory research is conducted at the beginning of development. Sprints have a fixed duration of 1 week. Each sprint begins and ends with a re-estimation and evaluation.	Sprint Human Centered Design	(Ardito et al., 2017)	**A12**
Inventive Problem Solving & Agile: They are integrating project-level innovations of so-called invention problem solutions into agile methods such as Scrum, transforming development teams to leverage their expertise in information technology to influence the overall solution architecture to maximize profits. As a result, innovative and efficient solutions are recommended and implemented.	TRIZ Scrum	(Fulbright, 2013)	**A13**
LEAN & Agile: This paper proposes a production control system built on lean and agile strategies to react to changes in the composition of the product mix by reassigning work elements in each workstation. Performance is analyzed under conditions of volatile changes in the product mix by experimental simulation.	LEAN	(Takahashi et al., 2007	**A14**
Lean six sigma & agile: This article combines six sigma, lean and agile development. Lean development and six sigma focus on performance improvement, and agile development focuses on rapid development. Their combination helps to quickly deliver high performance products to customers.	Lean six sigma	(Badwe and Erkan, 2018)	**A15**
Product family modelling & Agile: This paper describes structural modeling using two methods (called Kumbang and Agilefant). The integration aims towards advancing product development governance by contributing technology in terms of concepts and even automated tool support for preparation, monitoring, and commanding the development activity for various stakeholders.	Software Product Family Engineering	(Raatikainen et al., 2008)	**A16**

Quantitative Scheduling & Agile: They combine mathematical programming with agile development. This method can be used to generate schedules as a benchmark for agile iterative development. This approach is an effective complement to agile project management, especially if the budget is limited.	Daily Scrum Quantitative Scheduling	(Jahr, 2014)	**A17**
Rational Unified Professing & Agile: In proposed hybrid approach, 9 RUP disciplines have been reduced to 7 disciplines after considering suitable mapping with Scrum. The disciplines utilization in each phase may be different for iterations. Sprint plan meeting, daily scrum meeting and sprint review meeting can easily be embedded into elaboration and construction phases. Each sprint boosts with a sprint plan meeting and finalizes with the sprint review meeting.	Rational Unified Professing	(Tanveer, 2016)	**A18**
Risk Management & Agile: This paper proposes a risk management model that can be applied to the agile development process. This model defines security requirements, high-risk feature lists, identification functions, response functions, and monitoring functions. The model revolves around a risk board that is used and updated throughout the software development cycle.	Risk Management	(Ylimannela, 2013)	**A19**
Scrumban is a mix of scrum and Kanban. It can improve the applicability and versatility of product manufacturing and support the companies that care. This method is mainly used for fast start-up of dynamic environments.	Kanban Daily Scrum User-stories	(Matthies, 2018)	**A20**
Scrumconix uses the project managing framework of Scrum, and part of the software engineering guide of ICONIX. It is basically composed of two parts: Sprint Zero and Sprint One to N. Sprint Zero includes the following points: Overall requirements gathering; Domain and use case modelling; Reference software architecture; Overall project estimation; GUIs refinement; Sprint planning. The Scrum activities are unchanged.	Daily Scrum Sprint ICONIX	(Portela and Borrego, 2016)	**A21**
Security & Agile: Security assessments have been added to each of the agile processes to ensure security without changing the agility.	Sec. back-log	(Ghani et al., 2014)	**A22**
SEMP& Agile: A Systems Engineering Management Plan is a management method for contractors' systems engineering. A new hybrid approach that combines the principles of agility with the principle of SEMP to define the most critical role of the Product Owner (PO). He/she should understand the needs and expectations of the customer, understand the technical terminology, and be able to make individual decisions simultaneously. The development team is implemented in a similar way to the standard agile method.	Daily Scrum SEMP	(Lom et al., 2016)	**A23**
User-centered agile: The integration of User-centered design (UCD) and agile is mainly in the following parts: Little Design Up Front; Prototyping; User Stories; User testing; Inspection evaluation; One sprint ahead. This combination is based on the following principles: Separate Product Discovery and Product Creation; Iterative and Incremental Design and Development; Parallel Interwoven Creation Tracks; Continuous Stakeholder Involvement; Artifact-Mediated Communication.	Daily Scrum Demo User-centered design	(Aguilar and Zapata, 2017)	**A24**
XP & Throwaway Prototyping: Need to build a GUI prototype. The GUI prototype will be used for the final system. If the end-users are not satisfied with the GUI, the developers will go back to the analysis, design and implementation phase to come out with a new GUI and the process goes on and on until they are satisfied.	XP Throwaway Prototyping	(Alqudah and Abdulsalam, 2013)	**A25**

The Process of Hybrid Development Approaches: We found that most of the reviewed Hybrid Development Approaches were based on traditional development methods. That maybe since conventional development methods can be considered as a stable process, but practitioners maintain that this stability will make the development process rigid. Introducing agility to the processes is intended to make those processes more flexible. Scrum and Extreme Programming are the most usually used agile elements that were integrated into traditional methods. These outcomes are not extraordinary, since surveys dealing with agile approaches recognized Scrum and Extreme Programming as the most popular agile elements. However, regarding some Hybrid Development Approaches it was not specified which agile methods were used. The most common benefit of agile development is that customers can quickly see a prototype of a product and a validation can be made, because Scrum focuses on managing iterative development, and Extreme Programming focuses on incremental progress. Both bring higher productivity. Other purposes are optimizations for agile

development, such as reduce risk during the development process. Another example is establishing catalogue management for agile development because agile development focuses on the communication within teams but ignores the importance of documents. Agile development is generally considered to be suitable for small teams or small projects. To make its application more extensive, some articles propose optimization methods that can be applied to, for example, global development, or large projects and multinational corporation.

The Aims of Hybrid Development Approaches: Keeping the customer involved and actively collaborated in the development process, is most often mentioned aim. Next is trying to improve the cost-efficiency of the project and then accelerating development process. There are some other ones, like increases adaptability and universality for product manufacturing.

The Application Domain: The most familiar domain for Hybrid Development Approaches was Software development. Other fields include Manufacturing, web applications, game development, and telecommunication. The most of approaches were targeting a single domain. Nevertheless, few studies explicitly stated that they could be implemented more-ordinarily for large size projects such as **A18**.

5. Benefits and challenges of the identified approaches

Within the identified contributions, the benefits of combining plan-driven and agile approaches were reported in many aspects. Figure 2 displays the coverage of hybrid approaches with regard to the advantages identified by Schmidt et al. (2019) that companies hope to gain from integrating agile elements into their processes. It was found that only the approaches marked in bold were sufficiently empirical to make robust statements about the benefits of the approaches. For the remaining approaches, the classification is based on the literature - which in turn is partly based on assumptions.

Figure 2. Overview of the benefits of Hybrid Development Approaches

We made the assignment on the basis of the identified literature. Besides the aspect, that it is difficult to quantify the communication improvements which came from the adoption of the respective hybrid model this benefit is one of the most discussed. **A3, A5, A9, A11, A18, A21, A23,** and **A24** all claim that communication in the team has improved because of the agile elements. **A5** assumes that excellent team communication will bring more control and better management of project progress. **A11** mentioned that one of the biggest problems facing global multinational companies in the software development process is the communication of staff in different regions. Agile development practices have received a lot of attention for their flexibility, this remarkable feature has also been retained in **A5, A6, A10, A12** and **A24**. **A5** explains this in the way that the development should be based on early feedback from customers and on important product features, rather than focus working on rigid plans. **A24** supposes that agile can enhance UCD through higher frequency usability assessments, enabling customers' early feedback to be better integrated into the product. In terms of productivity, **A3** noticed a threefold increase in code output. Similar to that aspect, **A15** observed an improvement in productivity. Here it was based on the shorter turnaround times due to continuous focus on the deliverables. **A3, A11**, **A14, A20** and **A22** mentioned reducing the risk in the project. For example, **A22** defines bad behaviour and risk, and what measures can be taken to reduce these risks. **A1, A3, A9, A12, A14, A18, A20,** and **A22** highlighted that the time to market for the product has been shortened.

Articles containing the following hybrid approaches mention their problems. **A3** mentioned that, lack of understanding in middle or top management and isolation of agile teams from the rest of organization are the most common challenges they find. **A20** mentions that agile development can accept quick feedback from customers to guarantee that valuable products are implemented as quickly as possible. But the agile development team's company does not have a flexible and dynamic structure. Instead, they mostly have cumbersome processes. Adopting agile development does not change the underlying corporate structure of the company itself. Many articles have this assumption, and all work is based on a short-cycle form of a small self-organizing team. This means that the usage of hybrid development in large companies or large projects is also a huge challenge. There are also several common challenges as described following: Frequent customer feedback request, conditional suitability for distributed development and chaotic approach in the project.

6. Discussion & conclusion

In this paper, we have presented details of a Systematic Literature Review on Hybrid Development Approaches. In total we have identified 1628 contributions by a search string and reduced them by the Systematic Literature Review to 101 papers published between 2003 and 2019. The main characteristics of Hybrid Development Approaches were investigated, and the observed benefits identified. The results show that there is still much confusion about what Hybrid Development Approaches are and how the two different approaches of agile and plan-driven development can be effectively integrated. While the research method used enabled us to identify 25 different approaches that are the result of a combination of agile and plan-driven development, we must admit that we could only find literature that explicitly aimed to create a hybrid development approach. Other literature that also presents hybrid approaches in development and does not call them hybrid has not been found by the search string. The search agency website undermines the objectivity of the search and retrieval process and biases the review process. Differences in site search capabilities mean that search strings must be adjusted or discarded completely; and related sites may be excluded unintentionally (lack of knowledge) or for other reasons (time / resource constraints). This means that a large number of related studies may be missed. In order to achieve objectivity, inclusion and exclusion criteria were used to screen potentially relevant studies. However, subjectivity is inevitable during the screening process, especially when there are fewer researchers. Nevertheless, the findings in our contribution are comprehensive at most. Our results indicate that Hybrid Development Approaches are still in their early stages. The paper will contribute in advancing the state of research of Hybrid Development Approaches and can be used by researchers to bridge the gap in this area. However, there are not many cases in which the real success we are expecting from applying hybrid development approaches in the industry is discussed. Besides, the applicable limitations or challenges of these new hybrid theories are lacking, and there is no in-depth evaluation of these hybrid methods. Furthermore, not every paper has an in-depth study of this, and we have reviewed

it as much as possible here. There are, however, many gaps in this part of the content. We have located that the addition of agile elements can have a significant favourable influence on the development process in terms of customer, value, productivity, and internal processes. So, the similarity of these hybrid methods is to use incremental and iterative to change the rigidity of traditional development. In summary, most of the reports are optimistic through the mixture of agile and traditional development. They believe that the incorporation of the two must be able to blend their respective advantages. With this contribution, we have created the basis for investigating precisely this belief. We present an overview of relevant hybrid approaches in the design of development processes. Future research projects will focus on the situation- and demand-oriented combination of agile and plan-driven elements in order to influence the degree of agility in the development process in a targeted manner. (Albers et al., 2019b) In addition, our findings can be used as a basis for in-depth research into the systematic minimization of challenges in the combination of agile and traditional development. This allows the approaches to be used in practice in a targeted and meaningful way and not as an end in itself.

References

Aguilar, M. and Zapata, C. (2017), "Integrating UCD and an agile methodology in the development of a mobile catalog of plants", *Advances in Intelligent Systems and Computing* 486, pp. 75-87. https://doi.org/10.1007/978-3-319-41685-48

Albers, A. et al. (2019a), "Agility and its Features in Mechatronic System Development: A Systematic Literature Review" *Proceedings of 30th ISPIM Innovation Conference, Forence, I, June 16-19, 2019*.

Albers, A. et al. (2019b), "A systematic approach to situation-adequate mechatronic system development by ASD - Agile Systems Design" *Procedia CIRP*, 84, pp. 1015-1022. https://doi.org/10.1016/j.procir.2019.03.312

Al-Azawi, R. et al. (2014), "Multi Agent Software Engineering (MaSE) and agile methodology for game development", In *14th Middle Eastern Simulation and Modelling Multiconference, MESM 2014 - 4th GAMEON-ARABIA Conference, GAMEON-ARABIA 2014*.

Alqudah, M. and Abdulsalam, A. (2013), "Implementing computer-aided language learning tool using hybrid agile method: A case study", In *Proceedings - 2013 International Conference on Informatics and Creative Multimedia, ICICM 2013*. https://doi.org/10.1109/ICICM.2013.37

Ardito, C. et al. (2017), "Integrating a SCRUM-Based Process with Human Centred Design: An Experience from an Action Research Study", In *Proceedings - 2017 IEEE/ACM 5th International Workshop on Conducting Empirical Studies in Industry, CESI 2017*. https://doi.org/10.1109/CESI.2017.7

Badwe, S. and Erkan, T. (2018), "A taxonomy of lean six sigma and agile methodologies used in software development", In *International Journal of Engineering Research and Technology* 11, pp. 725-754.

Bauschmann, M. and Ahnert, C. (2017), Vergleich von Web of Science und Scopus im Hinblick auf den Informationsbedarf an der TU Chemnitz, 2017.

Boehm, B. and Turner, R. (Eds.) (2004), "Balancing Agility and Discipline: A Guide for the Perplexed. Software Engineering Research and Applications", Springer Berlin Heidelberg. Berlin, Heidelberg https://doi.org/10.1007/978-3-540-24675-6_1

Boehm, B. and Turner, R. (2005), "Management challenges to implementing agile processes in traditional development organizations", In *IEEE Software*, Vol. 22 No. 5, pp. 30-39. https://doi.org/10.1109/MS.2005.129

Cooper, R.G. and Sommer, A.F. (2018), "Agile–Stage-Gate for Manufacturers: Changing the Way New Products Are Developed Integrating Agile project management methods into a Stage-Gate system offers both opportunities and challenges", *Research Technology Management*, Vol. 61 No. 2, pp. 17-26. https://doi.org/10.1080/08956308.2018.1421380

Doshi, V.P. and Patil, V. (2016), "Competitor driven development: Hybrid of extreme programming and feature driven reuse development", In *1st International Conference on Emerging Trends in Engineering, Technology and Science, ICETETS 2016 - Proceedings*. https://doi.org/10.1109/ICETETS.2016.7602985

Fulbright, R. (2013), "Incorporating innovation into iterative software development using the inventive problem solving methodology", In *International Journal of Innovation Science*, Vol. 5 No. 4, pp. 203-212. https://doi.org/10.1260/1757-2223.5.4.203

Ghani, I., Azham, Z. and Jeong, S.R. (2014), "Integrating software security into agile-Scrum method", In *KSII Transactions on Internet and Information Systems,* Vol. 8 No. 2, pp. 646-663. https://doi.org/10.3837/tiis.2014.02.019

Goevert, K. et al. (2019), "Interview Study on the Agile Development of Mechatronic Systems", *Proceedings of the 22nd International Conference on Engineering Design (ICED19),* Delft, The Netherlands, The Design Society, https://doi.org/10.1017/dsi.2019.235

Haidar, H., Kolp, M. and Wautelet, Y. (2017), "Agile product line engineering: The AgiFPL method", In *ICSOFT 2017 - Proceedings of the 12th International Conference on Software Technologies*. https://doi.org/10.5220/0006423902750285

Hanschke, S., Ernsting, J. and Kuchen, H. (2015), "Integrating agile software development and enterprise architecture management", In *Proceedings of the Annual Hawaii International Conference on System Sciences, March 2015*. https://doi.org/10.1109/HICSS.2015.492

Heimicke, J. et al. (2019), "Comparison of Existing Agile Approaches in the Context of Mechatronic System Development: Potentials and Limits in Implementation", *Proceedings of the 22nd International Conference on Engineering Design (ICED19)*, Delft, The Netherlands, Design Society (Chair), https://doi.org/10.1017/dsi.2019.226

Ivček, M. and Galinac Grbac, T. (2008), Aspects of quality assurance in global software development organization.

Jahr, M. (2014), "A hybrid approach to quantitative software project scheduling within agile frameworks", In *Project Management Journal,* Vol. 45 No. 3, pp. 35-45. https://doi.org/10.1002/pmj.21411

Kitchenham, B. and Charters, S. (2007), Guidelines for performing Systematic Literature Reviews in Software Engineering 2.

Kumar, M., Shukla, M. and Agarwal, S. (2014), "A hybrid approach of requirement engineering in agile software development", In *Proceedings - 2013 International Conference on Machine Intelligence Research and Advancement, ICMIRA 2013*. https://doi.org/10.1109/ICMIRA.2013.108

Lom, M., Pribyl, O. and Zelinka, T. (2016), "System engineering for smart cities-hybrid-agile approach in smart cities procurement", In *WMSCI 2016 - 20th World Multi-Conference on Systemics, Cybernetics and Informatics, Proceedings 2*.

Matthies, C. (2018), "Scrum2kanban: Integrating kanban and scrum in a university software engineering capstone course", In *Proceedings - International Conference on Software Engineering*. https://doi.org/10.1145/3194779.3194784

Mukhtar, M. et al. (2013), "A hybrid model for agile practices using case based reasoning", In *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*. https://doi.org/10.1109/ICSESS.2013.6615431

Neumann, F. (2012), *Mechatronic Product Development: Potentials, Challenges, Terminology*.

Portela, L.T. and Borrego, G. (2016), "Scrumconix: Agile and documented method to AGSD", In *Proceedings - 11th IEEE International Conference on Global Software Engineering, ICGSE 2016*. https://doi.org/10.1109/ICGSE.2016.39

Raatikainen, M. et al. (2008), "Integrating product family modeling with development management in agile methods", In *Proceedings - International Conference on Software Engineering*. https://doi.org/10.1145/1370720.1370728

Saleh, S.M., Huq, S.M. and Rahman, M.A. (2019), "Comparative Study within Scrum, Kanban, XP Focused on Their Practices", In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox'sBazar, Bangladesh, IEEE*, pp. 1-6. https://doi.org/10.1109/ECACE.2019.8679334

Schmidt, T.S. et al. (2019), Agile Development of Physical Products: An Empirical Study about Potentials, Transition and Applicability, Report, University of the German Federal Armed.

Schuh, G. et al. (2018), "Agile-waterfall hybrid product development in the manufacturing industry - Introducing guidelines for implementation of parallel use of the two models", In *IEEE International Conference on Industrial Engineering and Engineering Management, December 2017*. https://doi.org/10.1109/IEEM.2017.8289986

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

Sohaib, O. et al. (2019), "Integrating design thinking into extreme programming", In *Journal of Ambient Intelligence and Humanized Computing,* Vol. 10 No. 6, pp. 2485-2492. https://doi.org/10.1007/s12652-018-0932-y

Takahashi, K., Yokoyama, K. and Morikawa, K. (2007), "Integrating lean and agile strategies into the production control system for mixed-model production lines", In *IFIP International Federation for Information Processing,* Vol. 246, pp. 405-412. https://doi.org/10.1007/978-0-387-74157-448

Tanveer, M. (2016), "Agile for large scale projects - A hybrid approach", In *2015 National Software Engineering Conference, NSEC 2015*. https://doi.org/10.1109/NSEC.2015.7396338

West, D. (2011), Water-Scrum-Fall Is The Reality Of Agile For Most Organizations today.

Ylimannela, V. (2013), A model for risk management in agile software development.

Zhang, Y. and Patel, S. (2010), *Agile Model-Driven Development in Practice 28*. https://doi.org/10.1109/MS.2010.85