Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 119 (2023) 1188–1197

www.elsevier.com/locate/procedia

33rd CIRP Design Conference

Analyzing current Challenges on Scaled Agile Development of Physical Products

Marvin Michalides[a]*, Nikola Bursac[b], Simon Jakob Nicklas[a], Stefan Weiss[c], Kristin Paetzold[d]

[a]Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, Neubiberg 85579, Germany
[b]Trumpf SE+Co.KG, Johann-Mauss-Straße 2, Ditzingen 71264, Germany
[c]AGENSIS, Cuvilliesstraße 14, München 81679, Germany
[d]Technische Universität Dresden, George-Bähr-Straße 3c, Dresden 01059, Germany

* Corresponding author. Tel.: +49-089-6004-2813 ; fax: +49-089-6004-4426. *E-mail address:* marvin.michalides@unibw.de

Abstract

Agile development has successfully established itself for coping with VUCA conditions on a team level. Furthermore, in developing cyber-physical systems or mechatronic products, the collaboration of multiple teams is often necessary. This necessity to scale contrasts with the paradigm of agile development from which additional challenges arise. The study examines the current challenges in scaling agile product development, characterized by scale-related challenges and physical constraints. This paper provides an empirical investigation to address these challenges. This publication shows that product development's top challenges are synchronization, coordination, resource allocation, and dependencies between different organizational units. It is determined that the challenges in the non-scaled environment also apply to scaling; however, substantial differences in their dependencies exist. Furthermore, an attempt to explain differences in scaled agile software development using comparative analysis is given. It becomes apparent that challenges such as production setup or physical limitations are non-existent.

Keywords: scaled agile; agile product development; challenges; comparative analysis; empirical survey

1. Introduction

Agile methods are established as a standard approach within the software domain. The application of agile methods, as well as the interest, is continuously growing in various sectors of industrial product development [1, 2]. A direct adoption from software development to hardware development is not trivial, resulting in arising challenges [3]. Nevertheless, manufacturing companies attempt to use new development processes to bring more complex and innovative products to market in uncertain environments. Therefore, among other things, efforts have been made to derive practices of agile development by adapting them to product development specifics [4]. The *Agile Systems Design* approach provides a further situation- and demand-oriented use of agile methods to support the development process [5]. Nonetheless, larger companies and corporations are subject to challenges of scaling within product development in order to be able to develop complex integrative products [6]. Due to the considerable success in software development, agile methods are already scaled with the help of various scaling frameworks [7, 8]. However, scaling agile and applying the specifics of physical product development to multiple agile teams leads to limitations, obstacles, and challenges. Thence, this contribution aims to fill the research gap by investigating these challenges in more detail by conducting an empirical study. Thus, the current challenges of scaling agile development of physical products are presented according to their relevance. Furthermore, the challenges of scaling agile hardware

10.1016/j.procir.2023.02.188

development are compared to the outcome of an explorative literature review that considers scaled agile software development. The corresponding research questions are:

RQ1: What are the challenges that need to be addressed in scaling agile development of physical products?

RQ2: In which aspects do the challenges differ from Software development?

2. State of the Art

Across this work, the terms "development of physical product" and "hardware development" are used interchangeably. Both terms refer to products or systems that do not consist exclusively of software, such as cyber-physical systems or mechatronic products [9]. Agility answers to new environments in product development characterized by volatility, uncertainty, complexity, and ambiguity (VUCA) and reinterprets previously perceived as an obstacle to generating competitive advantage [10]. Originally designed for small teams, agile approaches operate in the "sweet spot", and tend to be for collocated teams developing non-critical systems under VUCA conditions [11]. While doing so, agile approaches are based on short iterative development cycles and incremental development, which are grounded on the values of the Manifesto for Agile Software Development [12, 13]. While the application of agile methods in software development is already established, adapting to hardware development is associated with difficulties [14, 15]. Ovesen justifies these difficulties as an impediment to adopting agile methods in the field of physical product development with the Constraints of Physicality in his work [16]. In the non-scaled environment, sufficient work has already been published and discussed considering the understanding, potentials and challenges of agile product development [13-15, 17-20]. In addition to these challenges provoked by the Constraints of Physicality there is a need for adequate scaling to realize larger development projects and products. In fact, a separate multidisciplinary team responsible for the entire product development process would certainly not be able to develop a larger cyber-physical system [6]. In addition, scaling agile is neither trivial in software development [21]. However, scaling frameworks already exist in the software development domain to address some of these challenges [22, 23]. Furthermore, software development has the advantage of developing virtual, not tangible, products, so technical dependencies are more limited than hardware development [6]. Publications postulate that some scaling approaches are industry agnostic [24], but practices exist that are typical and significant to product development. Nonetheless, they argue that scaling cannot be adequately implemented without guidance and adaptation [25]. To establish a theoretical foundation, Michalides et al. explored the purpose of scaling agile and the application of various agile scaling frameworks in product development in their publication. However, they are guilty of addressing the current challenges in scaling agile development [26]. This publication aims to close this considered research gap.

3. Research Design

In this work, we analyze challenges that should be addressed in scaling agile hardware development and distinguish them from software development in a separate step. Addressing these challenges is crucial for implementing change-processes. What is more, this also impacts on the ability to scale within the organization to increase agility. Figure 1 illustrates the overall research process. The investigation is based on two data sets to answer the research questions. On the one hand, we collected empirical data by conducting an online survey, covering quantitative and qualitative data collection and analysis methods. On the other hand, we obtained and reviewed the literature for additional challenges in the intended context. The survey contained 42 questions. In total, 128 practitioners from different industry sectors participated. The study's area of interest was limited to the German-speaking region, i.e., Switzerland, Austria, and Germany. Moreover, the participants of the survey were filtered by implementing criteria within the demographic part. The filtering is meant to identify those participants who were developing physical products and had experience dealing with agile development [27]. Additional information can be found in [27]. The research platform Scopus was used to conduct the exploratory literature search. We used the search string *TITLE-ABS-KEY (((challenge*) AND agil* AND scal* AND development*)) OR (TITLE-ABS-KEY ((challenge*) AND agil* AND (mechatron* OR hardware OR physical)))* to narrow down the literature to be reviewed. The search returned 1232 document results. For the sake of clarity, the claim of the literature search was explorative in nature. Therefore, to further narrow down the literature, the filter criterion of the number of publications of the authors was used to select results that published at least three or more publications. The focus in analyzing the results was on identifying challenges in the context of agile development of

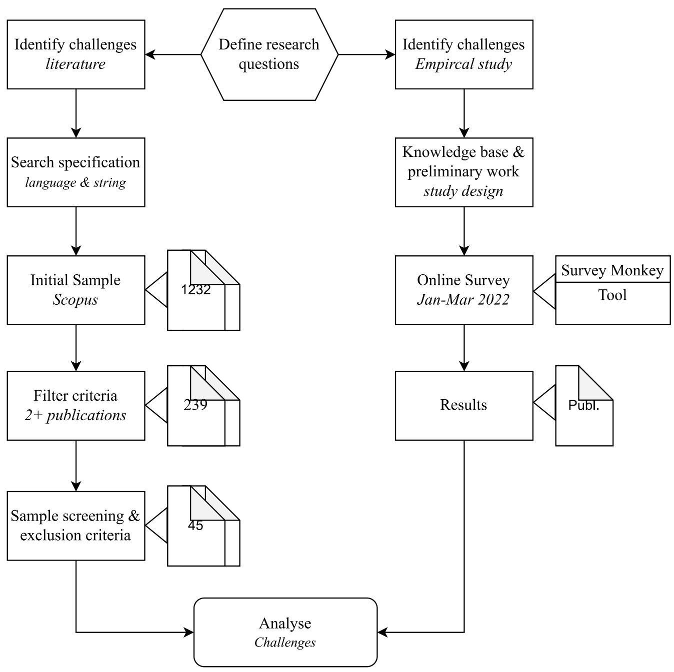

Figure 1: Visualization of the research process

physical products, including scaling, as well as possible referencing to the field of software development in the context of scaling. Additional sources were identified and evaluated in the primary source references. 45 results were evaluated as relevant. The relevant data from the literature review were extracted, subsumed, and compared in categories to highlight differences. Finally, results from literature and empirical study have been analyzed and compared.

4. Findings

This section presents and examines the current challenges that have been identified during our research. The results are summarized in Figure 2, Table 1 and further elaborated in the next section. In Figure 2, the challenges faced by the industry are listed in descending relevance. Each of the listed challenges was rated on a scale from 1 (no challenge) to 5 (very big challenge). The red line represents the mean value of the associated challenge. Dependencies in collaboration of agile and non-agile organizational units, Synchronization, Dependencies in collaboration of agile organizational units, and Coordination are considered to be the biggest challenges by far. Following Albers, organizational units could be understood as teams, departments or projects [28]. Since being rated fifth and sixth, resource allocation and communication have also been regarded as very challenging. On the contrary, there are no challenges that are not perceived challenging at all. Challenges were rated at least moderate. As mentioned in the research approach section, for further analysis, we mapped challenges between study results and related work, which are represented in Table 1. Table 1 lists the additional challenges identified in the literature review under the rated challenges from our study. Each of these has also been assigned a unique code to track whether the challenge originates from the empirical study (A-XX) or from the additionally performed exploratory literature review (B-XX). Therefore, Table 1 contrasts the related work with the results of this work. Three categories were distinguished: publications on scaling agile software development, publications on scaling agile development of physical products, and previous work from the area of agile development of physical products. In connection with the classified challenges, subsequent sources are represented coincident with the references.

The results put in evidence that compared to scaled agile software development, there are few publications on obstacles and challenges regarding scaling agile development of physical products. Some challenges have already been outlined, and others have not been actively stated as a problem area. For example, addressing the dependencies of agile organizational units has not occurred yet. Contrariwise, challenges have already been identified and classified in the development departments, which are also essential for scaling agile. A few of these challenges listed in Table 1. These are coordination, synchronization, communication, and potentially shippable increments. Furthermore, challenges seem to be non-existent in

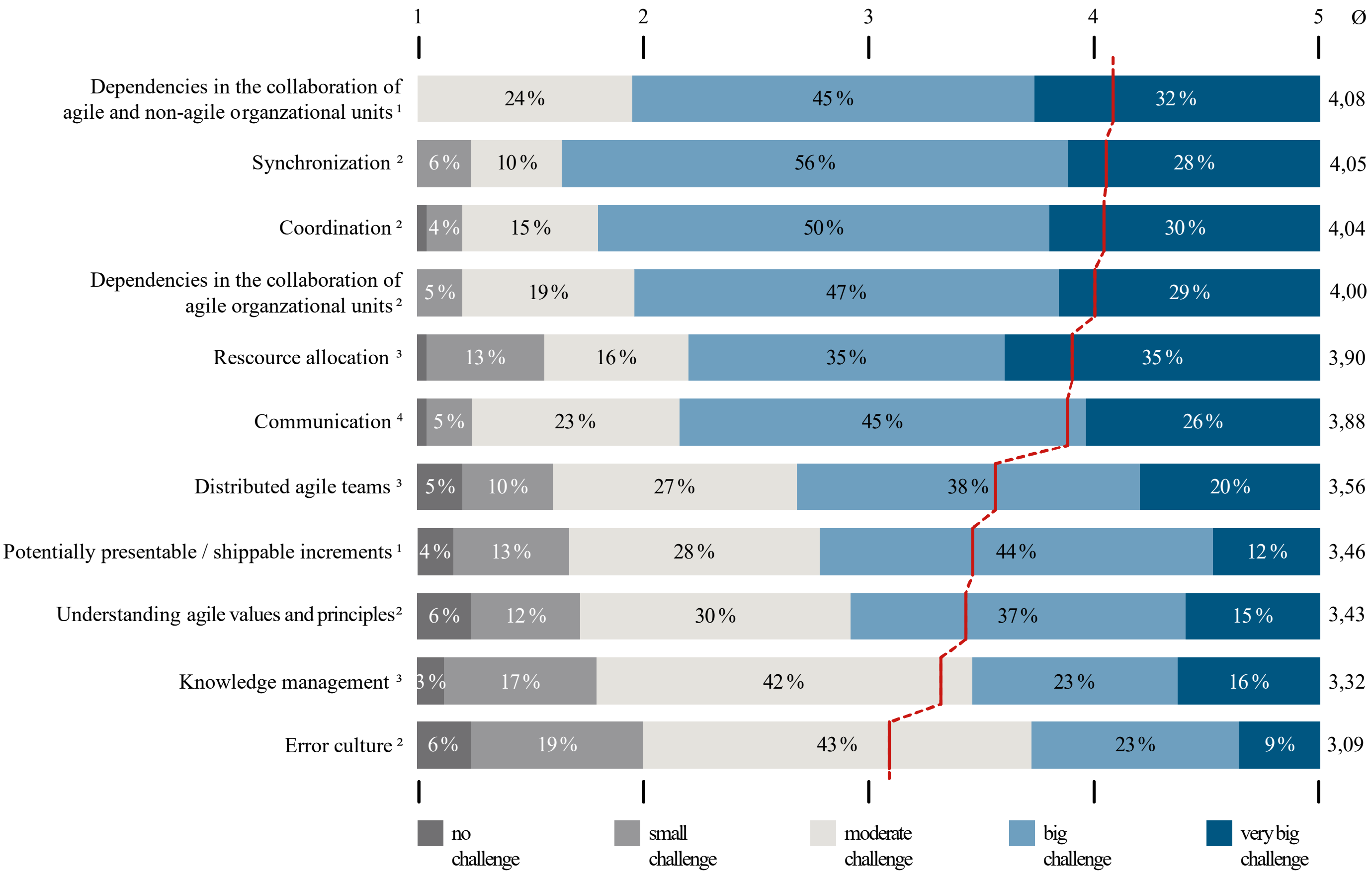

Figure 2: Evaluated challenges in scaling agile development of physical products, red line represents the mean values, percentages are rounded values, n=76[1];79[2];77[3];78[4]

Table 1: Mapping challenges between study results and related work

Code	Challenge type	Physical products related to scaling	Related to physical products	Software related to scaling
A-01	Dependencies in collaboration of agile and non-agile organizational units		[1], [13], [29], [30]	[8], [22], [31], [32], [33], [34]
A-02	Synchronization	[6]	[13], [29], [35]	[31], [36], [37]
A-03	Coordination	[6], [15], [24], [25]	[14], [29], [38]	[7], [21], [23], [31], [39], [40], [41], [42], [43], [44], [45], [46], [47]
A-04	Dependencies in collaboration of agile organizational units			[8], [44], [48]
A-05	Resource allocation		[16], [29]	[39], [46]
A-06	Communication	[15]	[13], [14], [18], [38], [49]	[23], [31], [36], [39], [41], [46]
A-07	Distributed agile teams		[1], [13], [38]	[32], [22], [23], [40], [41], [45], [46], [48], [50], [51]
A-08	Potentially presentable / shippable elements		[1], [13], [16], [29], [49], [52], [53]	
A-09	Understanding agile values and principles	[15], [24], [25]	[1], [3], [13], [16], [17], [18], [29], [53]	[2], [8], [21], [22], [23], [31], [32], [33], [36], [39], [41], [45], [46], [48], [51],
A-10	Knowledge management	[6]	[13], [27], [35]	[2], [7], [23], [36], [45]
A-11	Error culture		[14], [15]	[2], [31], [45], [48]
B-12	Regulatory compliance or government issue		[3], [13], [29], [49], [54]	[2], [42], [45], [48]
B-13	Insufficient training and education	[15]	[13], [17], [27]	[2], [22], [31], [33], [47]
B-14	Leadership			[2], [8]
B-15	Resistance to change		[13], [16], [17], [29], [30]	[2], [8], [21], [22], [31], [32], [34], [37], [41], [51], [55]
B-16	Inconsistent processes and practices	[6], [24], [25]	[13], [16], [17], [29], [30], [35], [49], [53],	[2], [37], [42], [51], [55]
B-17	Degree of physicality	[6]	[3], [13], [14], [16], [27], [29], [52]	
B-18	Organizational structure	[24], [25]	[13], [16], [18], [53]	[8], [21], [22], [23], [31], [33], [34], [37], [39], [41], [42], [43], [45], [47], [48], [50], [55]
B-19	Management support and sponsorship	[15]	[13], [17], [27], [30], [49]	[2], [8], [21], [22], [33], [41], [47], [48], [51], [55]
B-20	Quality assurance		[35]	[21], [23], [32], [33], [37], [51]
B-21	Testing	[24], [25]	[13], [29], [35]	[31], [33], [36], [45]
B-22	Production setup	[24], [25]		
B-23	Requirements Engineering		[52], [54]	[21], [22], [23], [31], [32], [33], [36], [37], [40], [41], [42], [43]
B-24	Product / software architecture	[15]	[3], [13], [14]	[7], [22], [41], [44], [47]
B-25	Customer collaboration	[6], [24], [25]	[3], [29], [35]	[2], [7], [33], [34], [36], [37], [51], [55]

software development. For instance, potentially shippable increments are not a challenge due to the virtual nature of the product. However, this is heavily relevant for tangible products, which consist of components from different company departments. The degree of physicality as a characteristic of tangible products also does not appear in the context of software development. In addition, the production setup is neither one of presented challenges. Last, in scaled agile software development, the challenges of coordination, distributed agile teams, resistance to change, requirements engineering, organization structure, and understanding agile values and principles have been frequently mentioned in the considered literature.

5. Discussion

The findings show several sources classified accordingly to the challenges categories, which are analyzed, compared, and interpreted regarding possible influence in practical application. The analysis is limited to the elements A01 - A11, they represent the challenges that were evaluated and rated in the conducted empirical study.

A01 – Dependencies in collaboration of agile and non-agile organizational units: Development departments consist of many different teams and units with varying degrees of dependency. This challenge category is a discriminator to capture the type of collaboration (traditional organizational unit / agile organizational unit) describing the correlation of collaboration/cooperation of agile and traditional organizational units. Different approaches also possibly different ways of working across divisions make it difficult for those groups to work together. Here, understanding the work context of other participating units is a formative factor in overcoming these challenges. Furthermore, different procedural approaches make interfaces more difficult to communicate and define. In principle, there is no significant difference compared to software development. The divergence is conveyed in the characteristics of the dependencies and constraints of physicality.

A02 – Synchronization: Within the synchronization process, there are differing levels of agile pervasiveness. Thus, these differences are particularly evident in the cross-cutting collaboration of agile teams. Especially in product development, the maturity level of agile teams is less consistent, which can lead to different boundary conditions in the synchronization process and hence to obstacles. Furthermore, differences can be argued concerning the type of synchronization between the various categories mentioned. Agile software development teams face different challenges than agile development teams in hardware development. The physicality of the products leads to further restrictions in the formulation of interfaces. The more complex the entire system, the greater the dependencies of its various subcomponents. This complexity, sequentially, implies dependencies between the collaborating agile teams. The number of external suppliers, stakeholders, and teams involved could also significantly influence the synchronization process. Furthermore, reacting to changes within a scaled context in which various dependencies exist is not trivial. It seems plausible that challenges in a non-scaled context also apply to the scaled context, when comparing agile development within the team with collaborating agile teams. However, they must be amplified, changed, and addressed differently and accordingly. The fundamental difference of synchronization between software development and hardware development manifests itself in subordinate artifacts and inherent activities to generate output.

A03 – Coordination: Collaboration distributes organizational, social, and technical issues that must be considered. On that account, this also includes the consideration of resources, resulting in different weights in the allocation and coordination process between individuals, teams, and areas. While scaling agile, the level of coordination of multiple teams and possible dependencies on the rest of the organization emerges. However, the coordination mechanisms among self-organized agile teams, as one principle of the *Manifesto for Agile Software Development*, represent a contradiction since dependencies on other teams could influence this agile principle. Therefore, scaling agile in connection with constraints of physicality leads to consequences of unsolved problems and coordination challenges. However, the coordination of several organizational units for the development of mechatronic products is a necessary factor if companies want to remain competitive using agile approaches. Thus, enabling agility in that manner is a crucial task. When it comes to fundamental understanding of coordination, differences in scaled agile software development compared to hardware development are less pronounced. Nonetheless, it can be assumed that additional mechanisms must be considered.

A04 – Dependencies in collaboration of agile organizational units: Compared to scaled agile software development, dependencies on other agile organizational units have not been addressed. This situation is not surprising since scaling has yet to receive much attention. These dependencies can be reflected in various ways in different areas. To name a few: task dependencies, causal dependencies, and preference-based dependencies. For instance, task dependencies could manifest themselves in necessary artifacts that other teams require to accomplish their task. So far, simulation results, technical drawings, or expert statements are common workarounds in that context. In the software domain, approaches such as designing processes and methodological support through frameworks are already being investigated. This challenge category is a discriminator to capture the nature of collaboration (agile organizational unit / agile organizational unit). Furthermore, this challenge presumably, also causally, includes the synchronization, communication, and coordination of these organizational units. The differences between software development and the development of physical products vary depending on the specific dependencies of collaborating organizational units.

A05 – Resource allocation: The application, prioritization, and allocation of resources, in terms of personnel and materials, both external and internal, is done differently. Different prioritization sequences of these resources are consequently common. Thus, finite resources potentially lead to undesirable side effects in a scaled context. Companies need to consider how allocations are made when agile developers are assigned to multiple projects or highly specialized skills are only available on a small scale. Consequently, breaking down silos within product development need to be discussed. The multidisciplinary of agile teams already counteracts the effect of silo mentality, although cross-collaboration leads to other challenges. Additionally, responsibilities must be examined, which received less attention in the context of organizations' hierarchical patterns. Differences in resource allocation could be related to the physicality of products. Moreover, this may be related to the challenge of properly breaking down the development task, which is more difficult in product development. Consequently, significantly more material resources must be considered compared to software development.

A06 – Communication: Agile developers can communicate directly through co-location within the team, which reduces the need for coordination and is beneficial in problem-solving processes. Communication structures in the agile sense are subsequently challenged, especially when direct communication is no longer possible. As a result of scaling, this could lead to restrictions and contradictions. Indirect communication flows complicate the exchange of information. Solutions to overcome communication barriers already exist through introducing and establishing remote software solutions. Development settings and environments differ significantly between product development and software development, so communication platforms are insufficient to solve collaboration problems during scaling agile. If communication is understood as the exchange of information, there are no significant differences between software development and hardware development. The difference stems from the domain-specific usage of signs and language. Consequently, the possibility exists that with the development of physical products, cross-domain communication is more complicated.

A07 – Distributed agile teams: Due to distributed teams at different locations, possibly across time zones, there are difficulties regarding feedback or agreements. This is not only a consequence of more difficult communication and collaboration but also due to cultural differences. Linguistic differences and possible contextual diversity, which could manifest in applying methods and processes, further complicates collaboration. Internationally collaborating teams also must develop under different political and legal constraints and laws, leading to obstacles and barriers. The most significant difference between hardware development and software development is the possibility of the physicality of an artifact. If artifacts are virtual, it could be supposed that the data and information transfer can run via a server. In that case, the differences are limited. However, if there are physical dependencies, this typically requires planning to minimize losses in terms of efficiency— for instance, possible efficiency losses resulting from delays in supply chain management.

A08 – Potentially presentable / shippable increments: This challenge does not seem to apply to software development, as the increments are provided in source code or functional software. Consequently, the delivered increments are used for validation by the customer. In the area of the development of physical products, this represents a significant challenge. There are still discrepancies in understanding how to apply agile methods, as they cannot explain how *Minimum Viable Products* can be produced cost-effectively in regular cycles. Nevertheless, validation by the customer in the further development process is of enormous importance. Taking Scrum into account, that could be an independent part of the product or a working prototype that the customer can validate. Nevertheless, these increments must be adequately defined so that the provided increment adds value to the customer. For this, the *Definition of Done* is a pragmatic way to meet evaluation benchmarks and set deadlines. Here solutions must be found to how the opportunity to update the projection of the desired project outcome can occur since a possible delivery within an iteration cycle seems unlikely. For this, other types of artifacts in the development process could contribute a significant role. In terms of scaling and collaboration of multiple agile teams, where product integration and synchronization are required, this is an even more significant challenge to solve.

A09 – Understanding agile values and principles: These challenges are often summarized under the term mindset. Mindset refers not only to the internalization of agile values and principles within an individual but also to the associated company culture. Concerning single individuals, it often becomes challenging to rethink plan-oriented approaches in the form of agile values. The collaboration between individuals is different, which on the one hand, can lead to more freedom for the development teams. However, on the other hand, it also leads to more responsibility within the teams. Already a challenge at a small scale, this internalized mindset is a prerequisite for scaling. The most critical part is harmonizing established approaches in product development on a micro-logical and macro-logical level. The fundamental logic of technical-physical development remains intact. Consequently, methodological assistance must be provided. In software development, there is no such difference in mindset regarding different levels of development because it is already assumed that developers know their craft. As a result, such adjustments are not necessary. Regarding scaling, mindset affects collaboration at interfaces due to the organizational structure, which can lead to problems between organizational units. Nevertheless, development departments are part of an organizational structure, so adjustments may be necessary to improve agile working and acting. Adjustments about moving away from hierarchical structures to lateral organizational structures are possible adjustment elements to promote and strengthen an agile mindset.

A10 – Knowledge management: In general, knowledge in companies is found in different granularity. The granularity differs in its diversity through the knowledge's relevance, topicality, depth, and breadth. Providing this knowledge is the essence of knowledge management [56]. In product development, the resource knowledge of the engineers involved is an essential guarantor for successful implementation within the product development process [57]. Consequently, the availability of knowledge must always be ensured as far as possible. If this explicit knowledge is not retrievable, this can lead to redundant work and, thus, to inefficiency. In agile development teams, there may be tacit knowledge that should be made explicit to the organization. In scaling agile, finding an effective way to provide implicit knowledge to different organization departments arises. In conformity with that, the processes, tools, and methods of knowledge management must also be addressed appropriately as part of scaling agile. However, an overload of documentation contradicts with the values of the *Manifesto for Agile Software Development*, although this does not preclude documentation. Regarding the manifest, it is about the balance that is inherently clarified. Nevertheless, solutions are necessary that address these challenges appropriately. Although knowledge differs in each development domain, the knowledge management challenges are similar. As a result, the differences between developing cyber-physical products and software development are limited.

A11 – Error Culture: A positive error culture requires transparency, openness, respect, and courage to establish a culture that should lead to constructiveness and improvement. Risks and failures in development are accepted in a solution-oriented approach, so an open exchange between developers can occur. Suppose Scrum is applied as one of the existing agile methods. In that case, different feasible solutions are explicated within a sprint. The customer validates the result of the increment. Therefore, the feedback will not necessarily be positive. Hence, it would be counterproductive to communicate any bad decisions within the team disrespectfully. Error tolerance and acceptability are essential. This fact could change as a consequence of the scaling. Accordingly, it must be ensured that there is no shifting of responsibility or prejudice against other teams in the scaled context. Within a team, the acceptance of errors is presumably higher because personal ties exist, and the shared vision is closer in perspective. Consequently, an informal relationship between developers is undoubtedly valued differently than a close personal relationship. Nonetheless, additional preferences regarding collaboration should be disregarded. In the context of this challenge, there is no difference in this challenge compared to scaled agile software development.

In summary, agile teams are multidisciplinary, self-organized, and work co-located in direct contact with team members and customers to remain competitive under volatile, uncertain, and complex environments. Scaling agile stems from a) scaling is mainly relevant for corporations and large enterprises to develop more complex systems, and b) these respective companies have already reached a certain transition level. Scaling processes complicate the practical implementation of the agile mindset, core concepts, and methods, which leads to challenges because the product structure reflects the organization's hierarchical relationships and communication structure [58]. This organizational structure stands partially in contrast to agile values. Benefits can be significantly reduced without sufficient addressing of the specified challenges. Some differences in scaling agile development vary between hardware and software development. For instance, differences can be disclosed in intensity, connectivity, and several different degrees of dependencies.

6. Conclusion, Outlook and Limitations

This paper analyses the current challenges associated with scaling agile in product development. Thus, the present work reveals empirically supported research bases for practitioners and researchers to develop a deeper understanding as a foundation for new research. Consequently, this contribution is intended to provide a starting point for addressing these challenges more intensively. Additionally, the empirical data offers a scored order of these challenges. Last, the existing literature is extended and supplemented by the challenges of scaling agile development of physical products.

The question of how the challenges in scaling agile manifest themselves in individual companies remains unanswered. Further research should investigate various interdependencies between challenge areas. Despite this, examining pre-existing challenges that need to be solved on a small scale, such as the presentable/shippable increments, and overcoming the constraints of physicality remains crucial. Nonetheless, data and information flows between teams dominate in the context of scaling, as the contribution shows. In this regard, an interesting approach might be to investigate leadership further and relate it to the challenges presented. This challenge could manifest itself inherently. For instance, the subject of leadership could change at various operational and strategic levels. At the operational level, postulated self-organization could be the first object of investigation for further research.

To address some of these challenges mentioned above, the publications of [28] and [5] offer first possible solutions for demand-oriented applications. Otherwise, in the software development domain, 147 practices are identified to facilitate application in the scaling context [59]. Following the work of [19], it is also important to investigate how holistic approaches contribute to scaling agile product development. To reduce the complexity of the interrelationships and dependencies between the challenges, the authors will conduct network analyses within the companies to provide initial points for solving these challenges.

Regarding the validity of the results, it should be noted that the empirical data were collected in the German-speaking region and are not representative for the entire manufacturing industry. Furthermore, the data were Likert data to achieve an appropriate comparability. Besides, we compared the results with the outcome from the literature review to classify our work. However, the subjective bias must still be mentioned in the interpretation, which we tried to reduce through group discussions.

References

[1] Schmidt, T.S., A. Atzberger, C. Gerling, J. Schrof, S. Weiss, and K. Paetzold, *Agile Development of Physical Products - An Empirical Study about Potentials, Transition and Applicability*. 2019, University of the German Federal Armed Forces:

München.

[2] VersionOneInc, *15th Annual State of Agile Report*. 2021, Digital.ai Software Inc.

[3] Heimicke, J., M. Niever, V. Zimmermann, M. Klippert, F. Marthaler, and A. Albers. *Comparison of Existing Agile Approaches in the Context of Mechatronic System Development: Potentials and Limits in Implementation*. in *Proceedings of the Design Society: International Conference on Engineering Design*. 2019.

[4] Schuh, G., C. Dölle, J. Kantelberg, and A. Menges, *Identification of Agile Mechanisms of Action As Basis for Agile Product Development*, in *Procedia CIRP Design Conference (28th CIRP)*. 2018: Nantes. p. 19 - 24.

[5] Heimicke, J., C. Czech, M. Schoeck, J. Mueller, S. Rapp, and A. Albers, *Introducing Agility into the Processes of Manufacturing Companies: A Method for Evaluating Success, Support and Applicability.* Proceedings of the Design Society, 2022. **2**: p. 2463-2472.

[6] Schrof, J., A. Atzberger, E. Papoutsis, and K. Paetzold, *Potential of Technological Enablement for Agile Automotive Product Development*, in *International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. 2019. p. 1-8.

[7] Dingsøyr, T., N.B. Moe, T.E. Fægri, and E.A. Seim, *Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation.* Empirical Software Engineering, 2017. **23**(1): p. 490-520.

[8] Putta, A., Ö. Uludağ, M. Paasivaara, and S.-L. Hong. *Benefits and Challenges of Adopting SAFe - An Empirical Survey*. in *Agile Processes in Software Engineering and Extreme Programming*. 2021.

[9] VDI, *VDI-Richtlinie 2206: Entwicklung cyber-physischer mechatronischer Systeme*. 2021, Beuth Verlag GmbH: Berlin.

[10] Schrof, J. and K. Paetzold, *Agile Produktentwicklung in einer zunehmend dynamischen Automobilwirtschaft: Potentiale und Grenzen*, in *41. Internationales Wiener Motorensymposium*. 2020: Wien. p. 362 - 376.

[11] Kruchten, P., *Contextualizing agile software development.* Journal of Software: Evolution and Process, 2013. **25**(4): p. 351-361.

[12] Beck, K.S., Ken, J. Sutherland, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, D. Thomas, S. Mellor, and R. Martin, *Manifesto for Agile Software Development*. 2001.

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

[14] Nicklas, S., M. Michalides, A. Atzberger, S. Weiss, and K. Paetzold, *Agile Entwicklung physischer Produkte - Eine Studie zum Stand in der industriellen Praxis während der COVID-19-Pandemie*. 2021, University of the German Federal Armed Forces: München.

[15] Atzberger, A., S. Nicklas, J. Schrof, S. Weiss, and K. Paetzold, *Agile Entwicklung physischer Produkte - Eine Studie zum aktuellen Stand in der industriellen Praxis*. 2020, University of the German Federal Armed Forces: München.

[16] Ovesen, N., *The Challgenes of Becoming Agile*. 2012, Aalborg University: Dänemark.

[17] Feldmüller, D., *Usage of agile practices in Mechatronics System Design - Potentials, Challenges and Actual Surveys.* 19th International Conference on Research and Education in Mechatronics (REM, 2018.

[18] Heimicke, J., M. Krüger, G.-L. Ng, N. Bursac, and A. Albers. *Agile Development in different Organizational Structures: Potentials and Challenges*. in *ISPIM Connects Global 2020: Celebrating the World of Innovation. Ed.: I. Bitran*. 2020. Online.

[19] Rößler, L. and K. Gericke, *Analysing Paradigms for Managing Product Development: Conventional, Agile and Hybrid Approaches.* Proceedings of the Design Society, 2022. **2**: p. 263-272.

[20] Albers, A., J. Heimicke, J. Müller, and M. Spadinger, *Agility and its Features in Mechatronic System Development: A Systematic Literature Review*, in *ISPIM Innovation Conference (30th ISPIM)*. 2019: Florenz.

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

[22] Uludağ, Ö., A. Putta, M. Paasivaara, and F. Matthes. *Evolution of the Agile Scaling Frameworks*. in *Agile Processes in Software Engineering and Extreme Programming*. 2021.

[23] Uludağ, Ö., M. Kleehaus, C. Caprano, and F. Matthes, *Identifying and Structuring Challenges in Large-Scale Agile Development Programs based on a Structured Literature Review*, in *21th Conference on Enterprise Distributed Object Computing (EDOC)*. 2019: Stockholm.

[24] Eklund, U. and C. Berger, *Scaling agile development in mechatronic organizations - a comparative case study*, in *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*. 2017. p. 173-182.

[25] Berger, C. and U. Eklund. *Expectations and Challenges from Scaling Agile in Mechatronics-Driven Companies – A Comparative Case Study*. in *16th International Conference, XP*. 2015. Helsinki.

[26] Michalides, M., N. Bursac, S.J. Nicklas, S. Weiss, and K. Paetzold, *Why Companies Scale Agile Development of Physical Products: An Empirical Study*, in *Design in the Era of Industry 4.0*. 2023, in press, Springer Nature: Bangalore.

[27] Michalides, M., S.J. Nicklas, S. Weiss, and K. Paetzold, *Agile Entwicklung physischer Produkte: Eine Studie zum aktuellen Stand in der industriellen*

Praxis. 2022, Universität der Bundeswehr München: Neubiberg.

[28] Albers, A., J. Heimicke, S. Trost, and M. Spadinger, *Alignment of the change to agile through method-supported evaluation of agile principles in physical product development.* Procedia CIRP, 2020. **91**: p. 600 - 614.

[29] Schmidt, T.S., A. Chahin, J. Kößler, and K. Paetzold, *Agile Development and the Constraints of Physicality: A Network Theory-based Cause-and-Effect Analysis*, in *21st International Conference on Engineering Design*. 2017: Vancouver. p. 199-208.

[30] Goevert, K., J. Heimicke, U. Lindemann, and A. Albers, *Interview Study on the Agile Development of Mechatronic Systems.* Proceedings of the Design Society: International Conference on Engineering Design, 2019. **1**(1): p. 2287-2296.

[31] Edison, H., X. Wang, and K. Conboy, *Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review.* IEEE Transactions on Software Engineering, 2021: p. 1-1.

[32] Kalenda, M., P. Hyna, and B. Rossi, *Scaling agile in large organizations: Practices, challenges, and success factors.* Journal of Software: Evolution and Process, 2018. **30**(10).

[33] Abrar, M.F., S. Ali, M.F. Majeed, S. Khan, M. Khan, H. Ullah, M.A. Khan, S. Baseer, and M. Asshad, *A framework for modeling structural association among De‐Motivators of scaling agile.* Journal of Software: Evolution and Process, 2021. **33**(8).

[34] Moe, N.B. and M. Mikalsen. *Large-Scale Agile Transformation: A Case Study of Transforming Business, Development and Operations*. 2020. Cham: Springer International Publishing.

[35] Goevert, K., A. Gökdemir, C. Peitz, and U. Lindemann, *Challenges of agile development implementation in mechatronic development processes*, in *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. 2017: Singapore.

[36] Kasauli, R., G. Liebel, E. Knauss, S. Gopakumar, and B. Kanagwa, *Requirements Engineering Challenges in Large-Scale Agile System Development*, in *2017 IEEE 25th International Requirements Engineering Conference (RE)*. 2017. p. 352-361.

[37] Kasauli, R., E. Knauss, J. Horkoff, G. Liebel, and F.G. de Oliveira Neto, *Requirements engineering challenges and practices in large-scale agile system development.* Journal of Systems and Software, 2021. **172**.

[38] Duehr, K., J. Heimicke, J. Breitschuh, M. Spadinger, D. Kopp, L. Haertenstein, and A. Albers. *Understanding Distributed Product Engineering: Dealing with Complexity for Situation- and Demand-Oriented Process Design*. in *29th CIRP Design 2019*. 2019. Portugal.

[39] Evbota, F., E. Knauss, and A. Sandberg. *Scaling up the Planning Game: Collaboration Challenges in Large-Scale Agile Product Development*. 2016. Cham: Springer International Publishing.

[40] Alsaqaf, W., M. Daneva, and R. Wieringa, *Agile Quality Requirements Engineering Challenges: First Results from a Case Study*, in *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 2017. p. 454-459.

[41] Paasivaara, M., B. Behm, C. Lassenius, and M. Hallikainen, *Large-scale agile transformation at Ericsson: a case study.* Empirical Software Engineering, 2018. **23**(5): p. 2550-2596.

[42] Steghöfer, J.-P., E. Knauss, J. Horkoff, and R. Wohlrab. *Challenges of Scaled Agile for Safety-Critical Systems*. 2019. Cham: Springer International Publishing.

[43] Cherinka, R., S. Foote, and J. Prezzama. *Lessons Learned in Adopting Agile Software Development at Enterprise Scale*. in *Proceedings of the 24th World Multi-Conference on Systemics, Cybernetics and Informatics*. 2020. Online.

[44] Salameh, A. and J.M. Bass, *An architecture governance approach for Agile development by tailoring the Spotify model.* Ai & Society, 2021. **37**(2): p. 761-780.

[45] Nägele, S., J.-P. Watzelt, and F. Matthes. *Investigating the Current State of Security in Large-Scale Agile Development*. in *Agile Processes in Software Engineering and Extreme Programming*. 2022.

[46] Kaur, K., A. Jajoo, and Manisha, *Applying Agile Methodologies in Industry Projects: Benefits and Challenges*, in *2015 International Conference on Computing Communication Control and Automation*. 2015. p. 832-836.

[47] Shameem, M., R.R. Kumar, M. Nadeem, and A.A. Khan, *Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process.* Applied Soft Computing, 2020. **90**.

[48] Gregory, P., L. Barroca, H. Sharp, and K. Taylor, *The Challenges That Challenge: Engaging With Agile Practitioners' Concerns.* Information and Software Technology, 2016. **75**: p. 26-38.

[49] Cooper, R.G. and A.F. Sommer, *Agile-Stage-Gate: New idea-to-launch method for manufactured new products is faster, more responsive.* Industrial Marketing Management, 2016. **59**: p. 167-180.

[50] Tarhan, A. and N. Ozkan, *Evaluation of Scrum-based Agile Scaling Models for Causes of Scalability Challenges*, in *Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering*. 2020. p. 365-373.

[51] Putta, A., M. Paasivaara, and C. Lassenius. *Benefits and Challenges of Adopting the Scaled Agile Framework (SAFe): Preliminary Results from a Multivocal Literature Review*. 2018. Cham: Springer International Publishing.

[52] Drutchas, J. and S. Eppinger, *Guidance on Application of Agile in Combined Hardware and Software Development Projects.* Proceedings of the Design Society, 2022. **2**: p. 151-160.

[53] Zasa, F.P., A. Patrucco, and E. Pellizzoni, *Managing the Hybrid Organization: How Can Agile and*

Traditional Project Management Coexist? Research-Technology Management, 2021. **64**(1): p. 54-63.

[54] Ovesen, N. and A.F. Sommer. *Scrum in the Traditional Development Organization: Adapting to the Legacy*. 2015. Berlin, Heidelberg: Springer Berlin Heidelberg.

[55] Conboy, K. and N. Carroll, *Implementing Large-Scale Agile Frameworks: Challenges and Recommendations.* IEEE Software, 2019. **36**(2): p. 44-50.

[56] North, K., *Die Wissenstreppe*, in *Wissensorientierte Unternehmensführung: Wissensmanagement gestalten*. 2016, Springer Fachmedien Wiesbaden: Wiesbaden. p. 33-65.

[57] Paetzold, K., *Data and Information Flow Design in Product Development*, in *Design Methodology for Future Products: Data Driven, Agile and Flexible*, D. Krause and E. Heyden, Editors. 2022, Springer International Publishing: Cham. p. 201-218.

[58] Conway, M.E., *How Do Committees Invent?* Datamation magazine, 1968. **14**: p. 28-31.

[59] Khan, R.A., M.F. Abrar, S. Baseer, M.F. Majeed, M. Usman, S. Ur Rahman, and Y.-Z. Cho, *Practices of Motivators in Adopting Agile Software Development at Large Scale Development Team from Management Perspective.* Electronics, 2021. **10**(19).