INTERNATIONAL DESIGN CONFERENCE – DESIGN 2020
https://doi.org/10.1017/dsd.2020.50

RELEVANCE OF PRODUCT INTEGRATION IN SCALED AGILE MECHATRONIC PRODUCT DEVELOPMENT

J. I. Schrof [1,✉], F. Rathert [2] and K. Paetzold [1]

[1] Bundeswehr University Munich, Germany, [2] Technische Hochschule Köln, Germany
✉ julian.schrof@unibw.de

Abstract

Industrial automotive development differs significantly from ideal agile conditions. Complex development structures, interlinkages between teams and non-functional physical dependencies between components result in agile constraints of scale and physicality. This qualitative study researches the influence of the product integration process on these constraints. The results show, that automotive integration characteristics such as duration, frequency, scope and transparency fail agile requirements and therefore cause constraints. Alternatives regarding IT and process design are discussed.

Keywords: agile product development, product integration, large-scale engineering systems, new product development, process improvement

1. Introduction

Interest and application of **agile product development** is continuously growing across different industries (Schmidt et al., 2019). The origin of this trend is an increasingly unpredictable business context. Digitalization is the connecting driver of this fundamental change in development dynamics (Kagermann, 2015). The VUCA acronym (volatility, uncertainty, complexity and ambiguity) summarizes its individual impacts on product development. Agile product development acknowledges the existence of uncertainty. It reinterprets previously as disturbance viewed change as opportunity and competitive advantage (Böhmer et al., 2015). Unpredictability and change are integrated as fundamental assumptions to the approach. Sequential product development methodologies like the waterfall system (Royce, 1970) or the VDI 2221 focus on predetermined development steps and detailed, long term planning to maximize efficiency. But far reaching plans are vulnerable to dynamic contexts, since the chance of change increases with planning horizon and level of detail. Agile methods on the other hand constantly adapt plans to minimize the offset to project reality. To do so agile product development is based on short iterative development cycles that allow fast feedback loops.

Benefits of agile software development such as flexibility, development speed and product quality have been shown in various case studies (Dybå and Dingsøyr, 2008). Due to its success the methodology was scaled to large projects and transferred to mechatronic product development (Furuhjelm et al., 2017). But application in mechatronic product development and the scaling to multiple teams resulted in agile constraints, caused by cooperation between teams. Compared to a single agile team cooperation between teams requires to repeatedly merge increments. This product integration step generates an additional feedback perspective: The system behaviour and the

components' influence on the system. Both are crucial for teams to integrate the system perspective into their component development.

This study researches the impact of product integration steps on constraints of physicality and scale in agile automotive development to understand the causes of agile constraint. The **aim of this study** is threefold. First, to summarize requirements of agile product development regarding the integration process. Second, to analyse the applicability of today's automotive integration process to support scaled agile development of mechatronic products. And third, to sketch integration alternatives based on IT and process adaptions. The corresponding **research question** is: *To what extent can constraints of agile automotive development be traced to the current integration system and what are alternatives?*

Even though there is a comprehensive research body on scaling agile product development (Dingsøyr and Moe, 2013) and transferring the methodology to mechatronic product development (Eklund and Berger, 2017) a **research gap** remains, regarding the integration process of scaled mechatronic product development and its unique characteristics. The relevance of this work is twofold. First, it increases applicability of agile product development to the automotive industry. Second, it expands the understanding of agile product development from a scientific point of view to avoid the guru problem (Janes and Succi, 2012) of the empirically emerged product development methodology.

2. State of the art

The **original agile approach from software development** is based on short iterative development cycles, incremental product development, cross-functional and self-organized teams and active end-user involvement which are summarized in the **Manifesto for Agile Software Development** (Beck and Beedle, 2001). Agile methods applicate and interpret these values and principles product- and context-specific (Abrahamsson et al., 2002). They present frameworks consisting of easy-applicable practices and shared principles intended to facilitate agile product development. Instead of long term plans and complex predictions, agile methods rely on continuous feedback to reduce divergence between project reality and planning. Consequently, an essential requirement for short iterative development cycles is fast validation and verification. Single cross-functional agile teams integrate testing into the team to realize direct feedback. Scaled approaches include several teams working on the same project. These teams rely on feedback of their component's behaviour as part of a system. In software development this led to **continuous product integration systems** (Wiest, 2010) based on automated product integration and testing procedures that constantly build and test new virtual product versions. Since developers get feedback on component and system level they can verify each adaption they implement straightaway. Examples of companies with elaborated integration system include Microsoft (Ebert and Paasivaara, 2017), and Saab (Furuhjelm et al., 2017).

Even though agile is established in software development a direct **transfer to automotive** development is not trivial. **Constraints to agile** automotive development can be categorized by **scale** (Sekitoleko et al., 2014; Uludag et al., 2018) and **physicality** (Ovesen, 2012). Their effects are approached with varying success by frameworks for scaled agile development (Alqudah and Razali, 2016), adapted agile principles (Conboy, 2010) and practices (Cao et al., 2004), restructured communication and complete organization conversions (Tyszkiewicz and Pawlak-wolanin, 2017). Scaled product development describes an increased project size that results in parallel, distributed and partitioned development (Eklund et al., 2014). Unlike software, mechatronic products need to be manufactured and are object to a network of physical dependencies between their components. Hence mechatronic product development requires additional design steps such as hardware verification, logistics and production, independent of project size. Even though both constraint fields have different causes, they result in dependencies between agile teams that have to cooperate on parallel and subsequent design steps. In scientific reports this inter team cooperation resulted in **coordination, communication and knowledge management difficulties** (Dikert et al., 2016; Dingsøyr and Moe, 2014; Sekitoleko et al., 2014; Uludag et al., 2018). All three categories are linked to the product integration system which includes component merge, system tests and information backflow to component development. The **automotive integration system** has to validate product relevance and verify both product functionalities and product properties. Product properties are caused by non-functional physical dependencies between components such as acoustics, thermal behaviour, vibrations

https://doi.org/10.1017/dsd.2020.50 Published online by Cambridge University Press

and crash behaviour which significantly increase the integration effort. Their complexity change the importance of system integration from necessary feedback to central value creation of OEMs. The integration process is implemented in month to yearlong V-model cycles that cause long duration total vehicle development.

3. Research design

To understand the connection between constraints to agile automotive development and the integration process the following **research structure** was chosen (Figure 1). First, differences between small scale software and large scale hardware agile product development applications were analysed. According to the results a model of requirements to agile automotive development was derived. Thereupon, qualitative data was collected via semi-structured interviews within one OEM to explore the current product integration process. Lastly, the industrial integration process was compared to the ideal model of requirements to understand and evaluate its impact on constraints to agile automotive development.

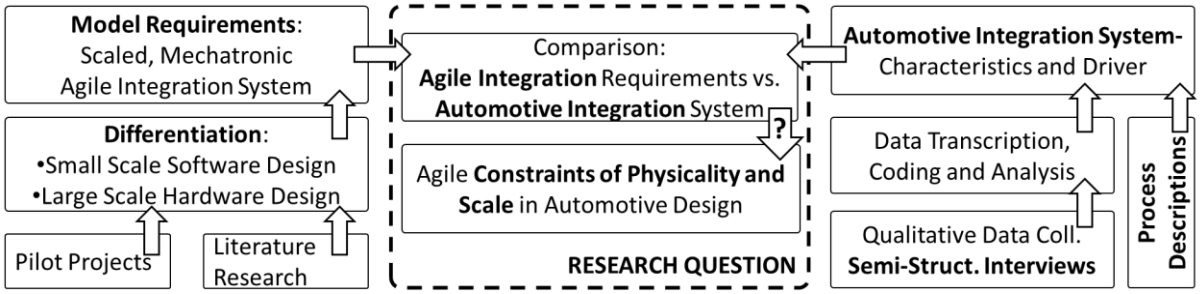

Figure 1. Research Design: A model of agile integration requirements is developed (left) and compared to automotive process descriptions (right) to understand causes agile of constraints

The **research focus** was the integration process of one automotive OEM to evaluate the requirements model. Specifically, the design-technique convergence in early product development was analysed. During this integration aesthetic design and technical restrictions are merged. It depends on actors from various organization units. Only the virtual part of the product integration was researched to avoid restrictions that are caused by production, hardware testing or logistics constraints but still deal with multi-physical dependencies between components on a total vehicle perspective. The relevant process steps are: component development, model build (data collection and system build), simulation, review and feedback (Figure 2). Aim of the simulations are verification of total vehicle functions and properties such as crash behaviour, fatigue strength and acoustics. While the interviews aimed at the model build steps, the preceding and subsequent process steps were included to understand dependencies and requirements of the complete process flow.

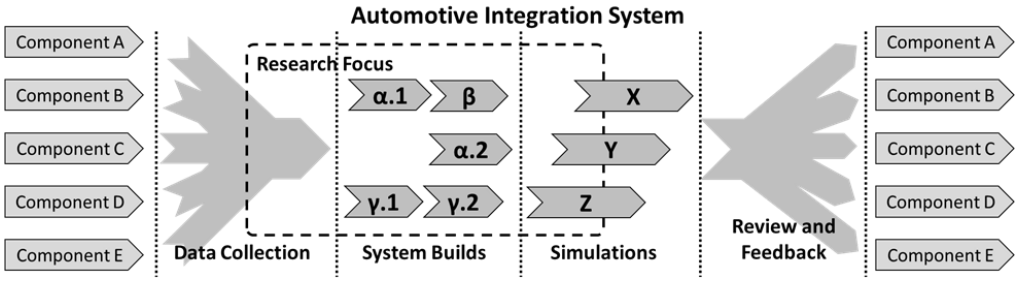

Figure 2. Virtual automotive design-technique integration; The sequential process exhibits dependencies between and within steps; Feedback is only provided after complete verification

The **research methodology** was commenced by a structured literature review (Vom Brocke et al., 2009) to support the research question and its scientific relevance. The theory building regarding differences between agile contexts and agile requirements of the scaled integration process of physical products were based on literature findings from case studies (Dybå and Dingsøyr, 2008), agile methods and the agile manifesto. Additionally, documentations from earlier agile pilot projects from

https://doi.org/10.1017/dsd.2020.50 Published online by Cambridge University Press

within the OEM were analysed. Shortcomings from practical applications were combined with theoretical descriptions to deduct the model. To validate the research question the generated model was compared to the status quo in industrial automotive product integration. To empirically collect data a **mixed method approach** was chosen. First, official OEM specific process documentation was analysed. This included general integration guidelines and specific design-technique convergence process descriptions. Second, **semi-structured interviews** (Bogner et al., 2014) with experts allowed to collect qualitative data. The combination of processual and interpretative data from the experts allowed to inductively differentiate a general representation of the real integration system and its short comings (Mayring, 2010). A qualitative approach was chosen since it supports understanding of a phenomena in its real-life context (Runeson and Höst, 2009).

A total of 14 interviews each 60 minutes long were conducted. Consistency between interview findings and preparatory exchanges show that the sample is sufficient for the chosen research focus. The interviews were recorded, transcribed and coded to allow statistic comparison and meaningful interpretation. Interviews and data analysis were conducted by different researchers. The experts were chosen according to functionality and component affiliation to account for department and product specific process particularities. One construction expert and five model build and simulation experts were interviewed on component level and six model build and simulation experts and two review experts were interviewed on total vehicle level. Four experts had plenty experience in agile development, eight had some experience and three were unfamiliar with the subject. The **interview structure** included the following steps. After a short introduction the first part of the interview was intended to confirm the overall process structures of the virtual integration phase to ensure relevance and scope of the interviews. A drawing of the process and relevant time marks were used as representations of these structures and the interviewee was asked whether she/he confirmed it. The second part was intended to draw a picture of the process from the experts' perspective. Process partners and their functions, time schedules, communica -tion paths, and cooperation models, integration of internal customers and general shortcomings were questioned to precisely model the real process. The third part focused on optimization potentials and asked about ideal team composition and typical technical or organizational problems during model build. In the fourth part the experts were asked to quantitatively and qualitatively assess value and applicability of agile practices to the integration process. These practices regarding coordination were selected from dominant agile methods and the agile manifesto, because earlier analysis had detected insufficient cooperation between stakeholders.

4. Results

The results section is divided into the description of the model of agile integration requirements and the interview results regarding the current design-technique integration process.

First, the deducted **model of agile integration requirements** is presented. Integration system characteristics vary significantly between single team and large scale agile applications. Regarding the agile sweet spot (Kruchten, 2013) of one compact team developing one software product responsibly, the check phase of the plan, do check and act phases of the Deming cycle (Moen and Norman, 2009) represents the integration system. Information from this phase affects all other phases and is distributed directly inside teams via internal informal communication (Pikkarainen et al., 2008). The team does not require formal communication because of the close collaboration. The same mechanism is not sufficient for large scale agile product development consisting of multiple teams. Assuming component specific testing is already included in component design, the following additional integration tasks are necessary. First **component merge**, the product components have to be merged virtually or assembled physically into a total vehicle. Product data packages have to be collected, interface specifications have to be checked and comprehensive total vehicle models have to be generated. The integration system requires product architecture knowledge, comprehensive model preparations, data accuracy and necessary IT infrastructure to support this step. Second **total vehicle verification and validation**, the technical product verification on a total vehicle level needs to be performed. Results have to be differentiated to component perspectives to give relevant feedback to teams. Product validation on as system level is even more relevant, since customers cannot segregate their product use onto different components. Still, relevant validation has to be derived to development

https://doi.org/10.1017/dsd.2020.50 Published online by Cambridge University Press

teams. Third **product documentation**, assuming independent teams the integration process has to support documentation on a system level to provide accessible and relevant information channels to teams. This includes total vehicle model data, testing strategies, procedures and results. Decision paths above team level have to be documented as well to allow retrospective analysis. Fourth **team coordination**, the integration system has to provide transparency regarding dependencies between components to coordinate teams and development. This transparency has to be updated dynamically according to iteration results and external changes. Fifth **communication between teams**, in single teams most of the formal communication (meetings…) can be substituted by direct informal communication (face to face…). This includes internal communication and implies most external communication. With a rising number of teams and functional and physical dependencies between components the need for information exchange and informal communication increases exponentially. Consequently, efficient external team communication has to be formalized in an agile multi team cooperation. Therefore the integration system has to provide documentation of verification and validation results, component specific feedback channels and access to total vehicle data models.

Further important variation between single team and scaled agile product development regarding integration are speed and frequency. Single teams accomplish their testing as a part of the iteration. Multiple teams can realize the same on component level but require additional time for system testing since component teams cannot include total vehicle testing. The delay between tested components and tested system generates either an **actuality gap** (Figure 2) between component and system development or decreasing efficiency with teams waiting for system feedback. The gap increases with system integration to component development ratio. Therefore scaled agile requires **frequent and short integration cycles**. In summary, product specific agile feedback mechanisms work well in single team applications. On a system of teams level an integration system is necessary. It has to feature the following characteristics: component merge, total vehicle verification and validation, product documentation, coordination and communication between teams as well as frequent and short integration cycles.

The second part of the results section presents the qualitative and quantitative feedback regarding the design-technique **integration process** from the expert interviews. The results are clustered into four parts. First, the experts' process descriptions are summarized. Frequency of changes to initial plans are presented and general differences between designed and actual process are reported. Second, specific short comings and recommended solutions are demonstrated. Third, the transferability and value of shorter integration length and smaller integration packages is presented. And fourth, the evaluations of the experts regarding the introduction of agile practices and fundamental process adaptions are listed.

Regarding the **status quo of integration process**, the experts agreed on the V-model based development and the six month time frame (see Figure 3). Planning horizons differed between few months up to a complete car development project with half the experts planning a whole car project development ahead and one expert planning less than six months ahead. Two thirds of the experts reported frequent external changes that contradicted original plans. Uncertainty and adjustments in initial premises were reported as well as a technology dependent change. Some changes could be anticipated other changes occurred unexpected. Their impact varied from local within teams to various other teams. The statements picture a dynamic development context requiring frequent change and adjustments.

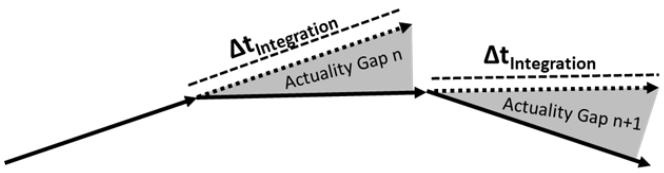

Figure 3. Actuality gaps caused by integration delays between component development with (full arrows) and without (dashed arrows) timely system feedback

Additionally, the network of process partners was described. All experts reported a high number of interfaces to process partners across organization silos and functionalities. Besides internal process partners a high participation of external service providers during the integration process was reported.

https://doi.org/10.1017/dsd.2020.50 Published online by Cambridge University Press

The integration system connects numerous specialists engaged in various projects. This division of labour results in a high number of process interfaces, handshakes and workflow interruptions. A significant part of the integration process crosses organizational units to combine functionalities which requires a high level of coordination. Besides the number of development interfaces the experts were asked to describe cooperation between process partners and consideration of subsequent process steps. Most expert described a loose contact and consultation only if necessary with process partners from different departments. Some reported coordination in committee meetings and others exchanged information in regular (weekly) meetings. No expert reported intensive cooperation and adjustment to internal customers. Driven by the high interlinkage between process partner only two experts were able to reach decisions independently. Responsibility was usually transferred to committees, supervisor or into teams of the same function, which decreased decision speed. The number of necessary decisions varied with the cooperation intensity.

Specific **process short comings and recommended solutions** are addressed in the following paragraph. Across all interviews problems regarding the model build step were reported. Their effect varied from additional workload for single teams up to complete job stopper for connected process partner systems. Insufficient component construction quality and inadequate data input were reported by all experts as relevant problems that affected their individual progress and overall process flow. Unpunctual and incomplete data delivery was reported the biggest problem to cause problems at handovers and required additional coordination effort. Unclear specifications, inconsistent data formats and problems with software tools were also reported. The actuality gap between construction and simulation was a major problem, since the construction experts required results and feedback from simulation to adapt their models. The experts also reported for the model build many manual steps and reviews that present a great share of the available time between comparatively fast simulations.

Regarding the option of **smaller integration packages** to reduce integration duration most experts replied that a complete total vehicle simulation was necessary to understand functional and physical effects that only arise on system levels. Since these effects have a significant influence on total vehicle characteristics, they are substantial for both component and total vehicle development perspectives. One third of the experts estimated that integration duration could be reduced in the present system.

Two enabler to **reduce total vehicle integration** length were mentioned: advanced IT infrastructure and a remodelled integration process. The new IT infrastructure system should include the following characteristics. Standardized data formats to manage system complexity and reduce specialization requirements. Automated data transformation and model generation to reduce manual data processing during development and at process interfaces. Process automatization to reduce manual tasks. The experts also declared the need for better process design covering coordination and communication between process steps and parallelization of model build and simulation.

Agile practices assessments are presented in the following paragraphs. The experts gave quantitative assessment on a scale from one to ten with one being no added value and ten being maximum added value (Figure 4). Additionally, qualitative explanations were supplemented.

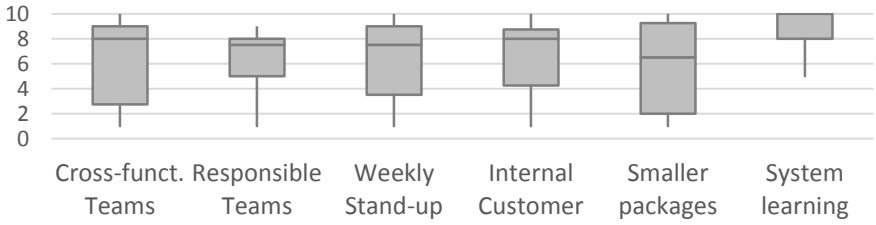

Figure 4. Evaluation of agile practices and principles in automotive product integration (Boxplot with 10 maximum and 1 no added value)

The first agile principle was **cross-functional teams** integrating relevant specializations in one team to reduce handovers and ease communication. A median assessment of 8.0 shows the importance in the experts' estimations. The scope of the answers varied from cross-functional teams in the own organization structure and even across departments. Some experts even added that despite their organizational placement they cooperate more intensively with their process partners from other departments. The

https://doi.org/10.1017/dsd.2020.50 Published online by Cambridge University Press

second agile principle was **responsible and self-organized teams**. A median of 7.5 confirms the positive assessment onto the overall integration process. Especially excessive bureaucracy was mentioned to slow development and increase complexity. The experts also stated that only a certain share of the compulsory meetings were necessary. They also stated that teams had to develop into this stage and instant change could not be expected. The third agile practice was short **weekly stand-up meetings**. A median of 7.5 shows the relevance of this efficient communication and coordination tool. But most experts added that the schedule had to be adapted to the rhythm of the project relevant results. In their cases simulations usually took longer than a single day and stand-ups only add value if new findings can be discussed. That's why ten experts agreed that meetings every day were not sensible, but two or three times a week or answering to demand were beneficial. The fourth agile principle was **customer integration** and aimed at the internal customer instead of the end user. The median value of 8.0 shows a high added value for this principle. Five experts stated they needed a more intensive cooperation with their internal customers and the others explained that they had sufficient connections already. The fifth agile principle was **smaller integration packages** and shorter integration iterations. A median of comparable low 6.5 shows that the experts had problems imagining smaller packages since they require the complete system feedback. A solution to this contradiction were repeated model updates that allow the integration of small packages into existing models and therefore enable fast total vehicle simulation. Seven experts share this statement and three confirmed that they develop accordingly. The last agile principle is **continuous system improvement** and a learning culture. A median of 10 shows the significance and value of this principles across the interviews. The majority demanded a new approach to ensure that mistakes are not repeated.

In summary, the interviews confirmed the expected process structure. Additionally, occurrence, frequency and impact of change in the plan driven process were uncovered. Complexity of process partners, communication flows and coordination structures were pictured. Shortcomings of the status quo such as insufficient data flow, slow decisions and unclear specifications were attested. The interviews elaborated that shorter integration cycles were impossible in the current system and recommended continuous model adaptions instead. To reduce integration length the experts saw two opportunities. The introduction of a new IT based integration system and an adapted process that outbalances current shortcomings. Agile practices were valued in general very beneficial.

5. Discussion

In this section the results are discussed regarding the applicability of the presented integration process to agile automotive product development. First, integration characteristics, system dynamics and differences between designed and real integration process are analysed. Second, a comparison between the requirements of agile product development to the real product integration process is drawn and the influence of the integration process on constraints to agile automotive development is discussed. Third, process evolution to reduce constraints and increase use in a more dynamic context are analysed.

Following the experts' descriptions of the integration process the subsequent process characteristics can be summarized. **Automotive integration** is based on long term, detailed planning between six months and few years ahead. Planning includes precise time schedules, information and data handover between process partners and role descriptions. Such detailed plans require precise and stable assumptions to be applicable. Contradictory, all experts reported unexpected change which required adaptions inconsistent with initial plans. **Frequency and occurrence of unexpected change** was most intensive in new product development areas. Change impact and linkage to other events is even intensified by a complex process network of stakeholder including external service provider. The number of process partners as well as insufficient communication at interfaces, increases process entanglement and reduces flexibility to change. Division of labour and missing mutual adjustment of in- and output between preceding and subsequent process partners creates a series of local optimizations which reduces overall information flow, cooperation and adaptability. This fragmented system was created to increase individual efficiency based on the assumption of a stable and predictable development context at the cost of transparency and flexibility. The reported lack of predictability leads to increasing change rates and therefore limit applicability of the current integration system. Instead, more adaptable integration structures are required in dynamic development contexts.

https://doi.org/10.1017/dsd.2020.50 Published online by Cambridge University Press

This imbalance has led to bottom-up process adaptions regarding feedback frequency. The long integration periods cause a significant **actuality gap** between component and system development. Its negative effects such as inconsistent data, incompatible interfaces and wasted development resources are caused by the inability to provide timely feedback on system levels to component development. Component teams are forced to continue their development without system feedback if they are to fulfil their development schedule. In this scenario probability of wasted resources and missed schedules increases with context dynamics and unexpected change. To reduce the actuality gap between system and component data, individual **component updates** are introduced to the total vehicle model. Other parts of the integration model remain constant to be able to start verification timely. This approach allows to reduce the actuality gap for limited subsystems and better deal with dynamic technologies.

The following section presents a **comparison between agile integration requirements and the current integration process**. The fundamental task of an integration system is to **merge components**. In the case of virtual product development this includes data collection, interface specification check, data format adjustments and standardization as well as the generation of total vehicle models for different system level testing. The analysed process fulfils all of these requirements to a certain extent. Even though some data gaps remain, they are not relevant for the simulations. Interface specification checks and data format adjustments rely mostly on manual steps which extends integration duration and affects quality. Considering **total vehicle verification** a complete set of simulations and other verification procedures are available and sufficient in the current integration process to verify total vehicle properties and functions. More challenging remains the differentiation of feedback on system verification to relevant components which is mostly discussed in meetings and not formalized so far. The **total vehicle validation** is more problematic, since the current integration process does not include repeated comprehensive customer validation. Customer integration strategies remain incomplete. **Documentation** of total vehicle models and simulation results for verification exist, but do not completely meet requirements of agile product development. Remaining data gaps have to be filled, actuality of component and system models have to be guaranteed. Also, information accessibility to developers is incomplete and lacks automatization. Additionally, decision trees have to be complemented to official documentation to avoid repetitive discussions in changing backgrounds. Insufficient transparency of dependencies between components results in inefficient **communication between teams** since exchange cannot be focused on relevant partners. Agile development requires an iteratively adapted transparency of component and team dependencies to guarantee **efficient coordination** of connected agile teams. Extreme division of labour and the corresponding number of interfaces are additional communication and coordination obstacles.

The described integration duration is as long as the component development time which results in a significant **actuality gap**. This contradicts agile integration requirements regarding speed and frequency of system feedback. Even though the longest simulation procedure lasts less than a week, sequential integration steps, process interface misfits, communication deficits, incomplete data, unfitting data models and a high share of manual labour cause long integration duration. The current integration length is a clear obstacle to agile automotive development. The integration duration also represents a lower limit to **integration frequency**. System integration is not required to match frequency of component testing. But the current one-digit number of cycles during the complete virtual product development is not sufficient for an increasingly dynamic context.

The presented imbalance between agile requirements to and the current form of product integration reveals that fundamental premises of agile product development philosophy are not supported. This conclusion allows to **answer the research question**: Even though automotive integration system supports some agile requirements it exhibits serious insufficiencies and therefore intensifies agile constraints of scale and physicality.

Integration system adaptions to reduce the constraints are presented below. Based on the agile requirements model and the interviews the integration system requires two central transformations to support agile automotive development. On one side, the IT system requires significant updates to improve data integrity, consistent data flow, fast data processing and data access. Current virtual integration logic was derived from sequential hardware development. It has to change to a continuous software

https://doi.org/10.1017/dsd.2020.50 Published online by Cambridge University Press

development process. On the other side, the structure and organization of the integration process have to be adapted to a more dynamic development context and a more powerful data backbone.
Even though both strategies are directly interlinked and depend on each other, this study focuses on processual adaptations. **Cross-functional integration teams responsible for complete verification process chains**, including data collection, data format standardization, model build and verification, could significantly reduce division of labour and thus cooperation complexity. The number of process steps, interfaces and process roles could be reduced considerably since teams would be responsible for process sections and not individuals for small process steps. Transparency of component dependencies would enable self-organized teams to contact each other directly and cooperate without consulting hierarchy. This would increase **external team communication efficiency** and reduce coordination efforts. Interaction between process partners requires a stronger cooperation across departments and silos. Better **understanding of process partners**' requirements allows to improve data and information flow and to reduce wasted resources on improper process handovers. Experts also underlined the importance of more frequent total vehicle verification feedback. C**ontinuous system improvement** has to become a new standard, since the current culture lacks bottom-up optimization. Retrospectives that address all integration process participants could realize optimization on a system level.

6. Conclusion

This study investigated the impact of product integration in scaled agile mechatronic product development. An interview series with 14 experts of one OEM was conducted to characterize the current integration systems in automotive development. The collected qualitative data was compared to a deducted model of agile requirements to automotive integration to analyse the connection between integration system and agile constraints of scale and physicality.
Complexity of automotive system integration is significantly higher than in software development because of additional development steps and physical dependencies between components that influence the system behaviour of the total vehicle. The current integration process relies on interlinked sequential process steps based on a detailed long term plan. Confronted with an increasingly dynamic development context this system is at the edge of feasibility. Unexpected change requires more flexible system integration.
Compared to scaled agile mechatronic system development the current integration system does not fulfil integration requirements completely. Even though component merge and system verification comply with agile specifications, system validation and integration documentation need further enhancement to support agile product development. Especially inter team coordination and communication rely on transparent system dependencies and more efficient formal communication channels. A major drawback of the presented system integration are long integration and testing cycles that prevent fast and frequent feedback on total vehicle perspective and cause actuality gaps between component and system development. These shortcomings are direct causes for agile constraints in automotive design.
Consequently, integration system evolution is a necessary prerequisite for agile automotive development. To increase integration speed and quality the following steps were proposed. Sequential process steps have to be condensed to avoid handovers and reduce coordination complexity. Parallel process paths have to be decoupled to avoid useless and harmful connections. Cross-functional teams manage responsibly complete integration process chains. Systematic learning is applied to continuously improve the integration process. Process and IT updates have to be developed in parallel to allow the IT system to answer processual requirements and vice versa. The lack of case studies from agile mechatronic product development with a focus on integration requires further field studies

References

Abrahamsson, A.P., Salo, O. and Ronkainen, J. (2002), *Agile Software Development Methods : Review and Analysis*, VTT Publication.

Alqudah, M. and Razali, R. (2016), "A Review of Scaling Agile Methods in Large Software Development", *International Journal on Advanced Science, Engineering and Information Technology*, Vol. 6 No. 6, p. 828.

Beck, K. and Beedle, M. (2001), "Manifesto for Agile Software Development", available at: http://agilemanifesto.org/history.html.

https://doi.org/10.1017/dsd.2020.50 Published online by Cambridge University Press

Bogner, A., Littig, B. and Menz, W. (2014), Qualitative Sozialforschung, Qualitative Sozialforschung, Oldenbourg Wissenschaftsverlag Verlag, München, available at: https://doi.org/10.1524/9783486717594.
Böhmer, A., Beckmann, A. and Lindemann, U. (2015), "Open Innovation Ecosystem - Makerspaces within an Agile Innovation Process", ISPIM Innovation Summit, No. December.
Cao, L. et al. (2004), "How extreme does extreme Programming have to be? Adapting XP practices to large-scale projects", *Proceedings of the Hawaii International Conference on System Sciences*, Vol. 37 No. February, pp. 1335-1344.
Conboy, K. (2010), "Method and Developer Characteristics for Effective Agile Method Tailoring : A Study of XP Expert Opinion", Vol. 20 No. 1, available at: https://doi.org/10.1145/1767751.1767753.
Dikert, K., Paasivaara, M. and Lassenius, C. (2016), "Challenges and success factors for large-scale agile transformations: A systematic literature review", *Journal of Systems and Software*, Elsevier Inc., Vol. 119,
Dingsøyr, T. and Moe, N.B. (2013), "Research challenges in large-scale agile software development", *ACM SIGSOFT Software Engineering Notes*, Vol. 38 No. 5, p. 38.
Dingsøyr, T. and Moe, N.B. (2014), "Towards Principles of Large-Scale Agile Development - A Summary of the Workshop at XP2014 and a Revised Research Agenda", *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation*, Vol. 199, pp. 1-8.
Dybå, T. and Dingsøyr, T. (2008), "Empirical studies of agile software development: A systematic review", *Information and Software Technology*, Vol. 50 No. 9-10, pp. 833-859.
Ebert, C. and Paasivaara, M. (2017), "Scaling Agile", *IEEE Software*, Vol. 34 No. 6, pp. 98-103.
Eklund, U. and Berger, C. (2017), "Scaling agile development in mechatronic organizations - A comparative case study", *Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track, ICSE-SEIP 2017*, pp. 173-182.
Eklund, U., Olsson, H.H. and Strøm, N.J. (2014), "Industrial challenges of scaling agile in mass-produced embedded systems", *Lecture Notes in Business Information Processing*, Vol. 199, pp. 30-42.
Furuhjelm, J. et al. (2017), "Owning the Sky with agile: Building a Jet Fighter Faster, Cheaper, Better with Scrum", pp. 1-4.
Janes, A. and Succi, G. (2012), "The dark side of agile software development", *SPLASH 2012: Onward! 2012 - Proceedings of the ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, pp. 215-227.
Kagermann, H. (2015), "Change Through Digitization—Value Creation in the Age of Industry 4.0", Management of Permanent Change, Springer Fachmedien Wiesbaden, Wiesbaden, pp. 23-45.
Kruchten, P. (2013), "Contextualizing agile software development", *Journal of Software: Evolution and Process*, Vol. 25 No. 4, pp. 351-361.
Mayring, P. (2010), *Qualitative Inhaltsanalyse - Grundlage Und Techniken*, 12th ed, BELTZ.
Moen, R. and Norman, C. (2009), "Evolution of the PDCA Cycle", *Society*, pp. 1-11.
Ovesen, N. (2012), "The Challenges of Becoming Agile: Implementing and Conducting SCRUM in Integrated Product Development", p. 200.
Pikkarainen, M. et al. (2008), "The impact of agile practices on communication in software development", *Empirical Software Engineering*, Vol. 13 No. 3, pp. 303-337.
Royce, D.W.W. (1970), "Managing the Development of large Software Systems", Ieee Wescon, No. August.
Runeson, P. and Höst, M. (2009), "Guidelines for conducting and reporting case study research in software engineering", pp. 131-164.
Schmidt, T.S. et al. (2019), Agile Development of Physical Products: An Empirical Study about Potentials, Transition and Applicability, Munich, available at: www.unibw.de/itpe.
Sekitoleko, N. et al. (2014), "Technical Dependency in Large-Scale Agile Software Development", *International Conference on Agile Software Development*, pp. 46-61.
Tyszkiewicz, R. and Pawlak-wolanin, A. (2017), "Agile Organization as a concept of production adjustment in the face of the crisis", *Production Engineering Archives Issn*, Vol. 15, pp. 19-22.
Uludag, O. et al. (2018), "Identifying and structuring challenges in large-scale agile development based on a structured literature review", *Proceedings - 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference, EDOC 2018*, pp. 191-197.
Vom Brocke, J. et al. (2009), "Reconstructing the Giant: on the Importance of Rigour in Documenting the Literature Search Process", *European Conference of Information Systems (ECIS)*, pp. 1-1.
Wiest, S. (2010). Continuous Integration Mit Hudson/Jenkins: Grundlagen Und Praxiswissen Für Einsteiger Und Umsteiger, dpunkt, Heidelberg.

https://doi.org/10.1017/dsd.2020.50 Published online by Cambridge University Press