Combining Agile Development and Software Product Lines in Automotive: Challenges and Recommendations

Philipp Hohl*, Michael Stupperich*, Jürgen Münch†, and Kurt Schneider‡
*Daimler AG, Research and Development, Wilhelm-Runge-Straße 11, 89013 Ulm, Germany
Email:{philipp.hohl, michael.stupperich}@daimler.com
†University of Reutlingen, Danziger Str. 6, 71034 Böblingen, Germany
Email: juergen.muench@reutlingen-university.de
‡Leibniz Universität Hannover, Software Engineering Group, Welfengarten 1, 30167 Hannover, Germany,
Email:kurt.schneider@inf.uni-hannover.de

***Abstract*—Software product lines (SPLs) are used throughout the automotive industry. SPLs help to manage the large number of variants and to improve quality by reuse. In order to develop high quality software faster, agile software development (ASD) practices are introduced. From both the research and the management point of view it is still not clear how these two approaches can be combined. We derive recommendations to combine ASD and SPLs based on challenges identified for an automotive-specific model. This study combines the outcome of a literature review and a qualitative interview study with 16 practitioners from the automotive domain. We evaluate the results and analyze the relationship between ASD and SPLs in the automotive domain. Furthermore, we derive recommendations to combine ASD and SPLs based on challenges identified in the automotive domain. This study identifies 86 individual challenges. Important challenges address supplier collaboration and faster software release cycles without loss of quality. The identified challenges and the derived recommendations show that the combination of ASD and SPL in the automotive industry is promising but not trivial. There is a need for an automotive-specific approach that combines ASD and SPL.**

***Keywords*— software product line, agile methods, automotive software development, software and system development process, ASPLA Model.**

1. Introduction

The automotive domain is recently in a disruptive change. E-Mobility, self-driving, and community owned cars are new technologies and upcoming market trends in the automotive domain [1]. A lot of innovation is nowadays addressed in software. Consequently, this leads to the fact that the amount of software in the car has increased exponentially in the last decades [2].

Software development in the automotive domain is a process with a high degree of reuse. Software Product Lines (SPLs) help to handle the large number of software variations, coordinate the worldwide software development, and increase software quality by the reuse strategy [3]. However, the development processes are too slow to keep pace with the fast changing market [4].

Since 2001, agile software development (ASD) methods promise a faster time-to-market with an increasing speed of learning. Using agile methods allows faster reaction on changed requirements from customers and market demands. The adoption of agile practices within the automotive domain is therefore a possible way to keep pace with fast changing market demands [5]. However, the combination of ASD and SPL development is assumed to be difficult [5]. Agile methods and practices are primarily designed for short development cycles in small development teams. One benefit of ASD is that the used methods emphasize collaboration and communication between developers as well as between the customer and developers [6]. For larger teams, agile methods are scaled up to approaches for large organizations, such as the Scaled Agile Framework (SAFe) or Scrum of Scrum (SoS). The scaled frameworks are aiming to maintain an open communication culture in and between development teams. However, this is challenging whenever the software development is distributed all over the world [7].

Various combinations of ASD and traditional development approaches exist within some parts of the automotive domain. These hybrid approaches are very specific for the context they are used in. This publication presents challenges for a hybrid approach to adapt ASD in the context of existing automotive SPLs for embedded software development. The identified challenges are based on the results of an interview study [8] and a literature review [9]. The findings help to integrate agile elements into regulated development domains by deriving recommendations from the identified challenges. The recommendations foster practices, which shall be taken into account. We limited the scope to automotive embedded software development because the automotive context has its special peculiarities, such as rigid quality and safety requirements [3], [10]. It is essential to maintain the functional safety of a vehicle frequently and continuously. Complex relationships between hardware and software development as well as in collaboration with supplies must be considered. Suppliers are often closely linked and integrated in the development process. Furthermore, specific testing conditions, like system or sub system integration of software and endurance tests in the real car must be considered [11].

The insights from this publication can be used by researchers and practitioners. The main goal of the derivation of recommendations is to support researchers in developing approaches to combine ASD and SPL in the automotive domain. These approaches can be techniques, methods, tools, processes, or organizational and transformation approaches. The recommendations might help researchers to get an overview on research demands with respect to the combination in the automotive domain. Furthermore, the recommendations presented in this article might act as a starting point for deriving recommendations for combining SPL and ASD in similar domains. This helps to analyze if current practices miss some recommendations or might need modifications. Process Engineers and other practitioners in the automotive domain that are involved in the design of development processes and organizational structures can use the recommendations as initial constraints that should be considered when combining ASD and SPLs. We expect that considering the recommendations will mitigate the risks associated with integrating ASD into SPLs. In summary, our contributions are as follows:

- Identify challenges that prevent a combination model of ASD and SPL within the automotive domain.
- Derive recommendations for an automotive hybrid approach from those challenges.
- Provide recommendations, which aspects are important when combining ASD and SPLs in the automotive domain.

The remainder of this paper is structured as follows. Section 2 describes related work. Section 3 reports the research approach. The research approach with the design, the data collection and the data analysis is described within this section. Section 4 summarizes the key findings, the contributions of the collected studies, as well as implications for practitioners. Section 5 discusses the findings. Finally, Section 6 concludes this publication.

2. Relation to existing theories and work

Da Silva et al. [12] conduct a systematic mapping study on agile software product lines. They identify agile practices which are suitable to combine ASD and SPLs such as scoping, requirements engineering and component development [12]. However, they determine that there does not exist a complete agile SPL approach. This issue is further examined by Díaz et al. [13]. They conduct a systematic literature review of experiences and practices on agile software product line engineering (APLE). They point out important challenges to set APLE into practice. They identify several barriers for the adoption of APLE such as the problem of traceability among the different SPL artifacts and the synchronization among product and platform teams [13]. However, Da Silva et al. [12] and Díaz et al. [13] do not focus on a specific domain. In contrast we conduct a literature review for combining ASD with SPLs in the automotive domain [9]. The literature review reveals there does not exist any automotive specific approach how to handle a combination of ASD and SPLs. Furthermore, we include related research areas, such as agile methods combined with embedded software development. With this approach, we aim to identify challenges for related research areas and to transfer the ideas and solutions to the automotive domain.

Holmström Olsson et al. [14] categorize their identified challenges into the single steps of the "stairway to heaven" model. The "stairway to heaven" represents the path of software development companies when moving towards agile practices. For the first two steps of the stairway, they identify challenges such as complex interfaces to corporate departments or the lack of test automation solutions and team discipline around test cases [14]. In addition, Gustavsson and Eklund [15] focus on the industrial challenges for scaling agile in mass-product embedded systems. They point out that in the embedded domain it is necessary to consider specific requirements, such as safety issues and the collaboration with other departments like electronics. In our literature review [9], we present various approaches and methods to overcome the identified challenges. We describe different agile methods and practices to combine ASD with a reuse strategy of software components. Agile methods, such as Scrum and XP are often described in the literature but are always adapted and tailored to the context of application.

Lindvall et al. [16] describe a project which utilizes the XP approach as a basis and extend it by auxiliary processes in order to define the scope and to adjust delivery time of the software in advance. They modify Scrum towards a new method called ScrumPL. In ScrumPL, the Scrum method gets tailored and software product line sub-processes are integrated into the process [17]. Iterative domain engineering as well as application engineering processes are performed in consecutive sprints [18]. Fitzgerald et al. [19] present an industry case study for scaling agile methods to regulated environments. The R-Scrum approach by Fitzgerald et al. [19] is the standard Scrum model extended by additional elements, artifacts and roles. Their approach of R-Scrum presents a directly usable framework. However, the framework must be adopted or further tailored as needed. This publication can serve as a starting point for a combination, but it does not consider the usage of a software product line. Another approach to merge SPLs and agile development is the Feature Driven Approach. This approach uses Feature Models [20], [21] and captures the initial view of the results of the commonality and variability analysis. With the feature orientation it is possible to blend the benefits of product line engineering with those of agility [22].

To combine ASD and SPL, various approaches exist in literature. None of them is intimately tailored to the automotive domain. In contrast to the presented publications, we are working on a combination model. Our contribution is to identify challenges in the literature review [9] and in the interview study [8] and extend the findings by deriving recommendations for an automotive specific combination model.

3. Research Approach

3.1. Research Question

This paper aims to answer the following research question: *What are the challenges and consequences for a combination of agile software development and software product lines in the automotive domain, according to published literature and real-life experience?*
To answer this question, two research questions are defined. The questions help to identify the hindering challenges for the automotive domain and help to overcome these challenges by defining recommendations.

RQ 1: *Which challenges can be identified for setting up a combination of ASD and SPLs in the automotive domain?*
The goal is to identify existing challenges, based on results from our literature review [9] and our interview study [8]. The challenges are separated, reviewed by another researcher, and entered into a requirements document in PTC Integrity[1]. We use PTC Integrity to maintain the dependencies between challenges and recommendations.

RQ 2: *Which recommendations can be identified for the combination of ASD and SPLs in the automotive domain?*
This question investigates recommendations which help to extend SPL development in the automotive domain by agile development methods. The recommendations are based on identified challenges and form a combination model to combine ASD and SPL in the automotive domain.

3.2. Research Design

In previous research, we have conducted a qualitative interview study [8] and a literature review [9]. Our work combines the results of these publications and extends them by recommendations to combine ASD and SPLs. Furthermore, we include the experience of Researcher 1 (P. Hohl) and Researcher 3 (M. Stupperich), due to their active involvement in automotive software development.
Interview Study [8]. The interview study is based on a qualitative survey. It is designed as an exploratory semi-structured interview, with an interview guide. The interview guide was tested in a pilot interview and adjusted to the problems which have arisen.
Literature Review [9]. The research process described by Petersen et al. [23] served as a basis for the literature review. Two researchers were involved in the searching process [9].

3.3. Data Collection

3.3.1. Interview Study. We selected the interview participants from employees of an OEM and one automotive consultant. The interviewee selection was based on two criteria: First, the interviewee should have a work experience of several years. The length of employment varied from 3 to 20 years, with an average working experience of 16 years. Second, the interviewee should already use agile practices for software development. To get a different point of view on the examined topic, we selected the following participants: Two managers, five process owners, two system architects, six software developers and one automotive consultant for agile development processes.

Researcher 1 conducted 14 face-to-face interviews as well as a group interview with two participants. Each interview took around one hour. The interview questions were initially defined in English and translated to the native language (German) of the interviewees. In consent with the interviewee, we recorded the interview and transcribed it verbatim for detailed analysis. All transcribed interview notes were managed using the reference management program Citavi.

3.3.2. Literature Review. As mentioned, two researchers were conducting the survey. Researcher 1 used the research process based on Petersen et al. [23]. Eight databases were selected for the mapping study: Scopus, Science Direct, IEEE, ACM DL, Springer Link, Wiley Online Library, World Scientific, and Google Scholar. With these databases, the researcher conducted a comprehensive search to avoid bias. As proposed by Peterson et al. [23], Researcher 1 constructed a search string based on the three main keywords "agile", "automotive", and "software product line". For the three keywords, synonyms and a corresponding word family were defined [9]. The search was conducted using titles, abstracts and keywords.

The literature review was extended by the use of the snowballing method. The "Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering" by Wohlin [24] were applied. Five iterations with forward and backward snowballing were performed.

3.3.3. Action Research. Researcher 1 and Researcher 2 are involved in an automotive embedded development process. The data collection is based on their every day's work and used to extend and verify the findings. The definition of the additional recommendations are based on our experience from managing a SPL for worldwide automotive embedded software. The recommendations were validated by a development team from the automotive domain.

3.4. Data Analysis

The identified challenges extracted from both studies were clustered into appropriate categories and concepts. This was done by Researcher 1, using concepts of Straussian Grounded Theory based on the classification of Stol et al. [25]. We used the three coding phases: open coding, axial coding, and selective coding [25]. The open coding breaks down the data analytically. This process step generates categories and concepts. The concepts were clustered and related to their subcategories in the axial coding. As a last step, central categories were defined in selective coding.

All clustered challenges were entered into the tool PTC Integrity Lifecycle Manager. PTC Integrity was selected

1. PTC Integrity is a software system lifecycle management and application lifecycle management platform used in the automotive domain.

because the authors are using the tool for requirements management and the management of the software development process. Furthermore, it is easier to handle reviews within the team since the developers are familiar with the tool.

A requirements management document was created in PTC Integrity and separated into two sections. One section for the challenges based on results from the literature review [9], and one section for the challenges determined by the results of the interview study [8]. Based on this, recommendations for a hybrid approach in the automotive domain could be derived. Furthermore, we introduced additional recommendations based on the experience of the development team. Each additional recommendation was traced towards a challenge to keep the hybrid model consistent.

3.5. Threats to Validity

According to Wohlin et al. [24], we address the four categories of validity concerns: Internal, external, conclusion and construct validity.

Data Extraction. Data extraction, identification and categorization of the challenges was done by the author. This could lead to bias and incorrectly presumed challenges. The tracing and derivation of recommendation could be biased and wrongly categorized. Missing links and traces within PTC Integrity are a threat to *conclusion validity*. To overcome this, the recommendation and traces were peer-reviewed by another researcher.

Researcher bias. The data extraction and coding was executed by Researcher 1. This could introduce bias due to misunderstanding, misinterpretation, wrong categorization and incorrectly set traces in PTC Integrity on the researcher side. Review steps with other researchers help to verify a sound research process. Furthermore, the active involvement into the field of interest of Researcher 1 could introduce bias. We reduced this bias by publishing the research and embedding the research into the scientific community. We validated our results in discussion rounds with several researchers.

4. Findings

4.1. Reseach Question 1

4.1.1. Interview Study. Based on results of the interview study [8], we identified several challenges. These challenges can be categorized into (1) organizational challenges for worldwide development, (2) challenges related to the coordination of the software product line, (3) challenges in adopting agile elements within the development process, (4) and challenges regarding automotive specific constraints.

Organizational challenges (1) for introducing agile elements to the existing SPLs are the complex coordination of different departments and the worldwide distribution of development teams. To manage worldwide distribution, big companies in the automotive domain are often organized in a complex hierarchical structure. Several management levels are involved in the coordination of the development. An identified challenge is that the managers within upper management levels do not want to lose any control. They are often addicted to the old paradigms of top-down management and it is hard to change them towards an open minded, agile way of developing software. They argue that the existing processes are not bad and still valid. The managers also considered it hard to plan and coordinate the development in an agile working environment. Therefore, the higher management levels stick to specific milestones in the processes, which foster the worldwide coordination and planning of the software development process. A synchronization of the software development with other processes like hardware, supplier involvement and in-house development is challenging. Working with suppliers gets even harder if the development is distributed worldwide. Co-located teams across the borders of countries were mentioned to be challenging. As it becomes apparent in the interviews, cross-cultural understanding is hard to achieve. Different cultures and different mindsets are likely hindering an agile development. A worldwide development leads to challenges in the communication within the team. Problems in communication are based on different time-zones, no face-to-face conversation and mistakes in translation.

To combine existing SPLs (2) and ASD (3) it is necessary to consider the software baseline. A lot of planning and coordination is necessary to maintain the SPL. It is necessary that the whole organizational structure is considered. The organization comprises a lot of different domains. Worldwide software development teams are responsible for technical systems, test and verification departments and other developments domains like hardware and mechanical. As mentioned by the interviewees, it is necessary to synchronize the development through all departments and through all hierarchy levels. The process to maintain and scope the common software parts requires a lot of communication. This communication was identified for slowing down the development by the interviewees. Communication channels within teams and across the organizational hierarchy are identified to be too slow [8]. The coordination among the SPL is challenging and it is unclear how agile development practices can be integrated into the process. One challenge mentioned in the study is to prevent a decomposition of the SPL. The SPL could decompose due too many branches in the software base line and a high amount of variation points within the software. It is necessary that the common software part is not neglected and maintained continuously. Handling different variants and freezing the functionality for several variants of the SPL independently is challenging. The developers mentioned that it is important to scope the functionality to maintain the SPL. Each affected variant must be identified because all variants shall work as expected and defined by standards. Within the SPL and the high amount of different software variants, a good and scalable test framework is essential. So far, it is unclear for the interviewees how far the automation of tests for the high amount of variants could be realized.

A benefit of ASD is to re-prioritize features, which should be implemented. However, the study revealed that the coordination of the software development hinders a faster pace.

The automotive domain (4) is a highly security and safety critical domain. It is highly challenging to be compliant to standards and the law (national and international) while developing a lot of different software variants. All participants mentioned that an increase of the development speed urges the validation of the software in the same pace. Maintaining standards like ISO 26262 and other restrictions given by the law are not disputable. Several challenges were mentioned regarding the software development process and the connected areas in the interview study. One participant mentioned that the automotive domain is a highly cost-driven business. The hardware is often designed to the minimum of acceptable performance and the minimum assembly of components on the circuit board. This does not mean that there is a reduction of quality or functionality, but it leads to the unpleasant effect that different variants for specific markets are compiled separately to fit the hardware. One solution is a modular software architecture which is capable to handle not foreseen features [8]. This modular architecture is a prerequisite for further progress, but often hard to put into practice. To get such a high modularity, it is necessary that the architecture is well chosen. One challenge is to maintain a good shaped architecture but to have the possibility to integrate not foreseen features into the software at the same time. Furthermore, it is necessary that all adoptions are always in relation to the selected hardware target and software variant. The study revealed that the coordination of the development is therefore a main challenge which must be addressed.

Overall, we identified 59 challenges based on the results of the interview study [8] for the integration of ASD within existing automotive SPLs. With the challenges, it is possible to derive recommendations and keep the trace to the corresponding challenge. An exemplary presentation of the challenges is presented in Table 1.

TABLE 1. EXEMPLARY PRESENTATION OF CHALLENGES FROM THE INTERVIEW STUDY.

ID	Category	Text
135873	Automotive	Maintaining the compliance to automotive standards, such as ISO26262 is seen as highly challenging.
140550	Automotive	It is unclear how each software release could be tested on real hardware in a car.
136178	SPL	Test coverage of a lot of different software variants is hard to keep high.
135878	SPL	It is a challenge to scale the test framework to test all variants within the SPL.
135880	SPL	It is unclear how far the automation of test could help in the process.

4.1.2. Literature Review. In the literature review [9], we approached the issue of ASD in combination with SPLs in the automotive domain from three different perspectives. The first perspective focused on existing ASD in the automotive domain. It examined if SPL development could be combined with already implemented agile processes. None of the solution approaches addressed the existence adoption of agile practices for the established SPLs in the automotive domain.

The second perspective focused on existing SPLs in the automotive domain and spot on the combination with ASD. This revealed that software product lines within the automotive domain are widely used. However, the publications on SPL in the automotive domain typically do not consider agile development approaches. A key finding for the literature review is that there exists no ready-made solution to combine agile software development with software product lines in the automotive domain [9].

The third perspective takes related research areas into account. Challenges and possible solutions from related research areas could be identified to overcome the research gap for agile SPLs in automotive. The published literature provides several sources of information on concepts for the combination of agile and product line concepts. These concepts are used as candidates for the adoption in the automotive domain, because they reveal helpful insights for an adaptation. In the literature study, several concrete methods and models are described that combine ASD and SPLs. Several best practices from related research areas could be identified. Pohjalainen [26] describes a bottom-up approach to combine product line engineering and formal modeling with agile software development. Hanssen [27] concludes that a combination makes the organization more flexible and thus capable of serving a volatile market with fast-changing technologies. Furthermore, this enables the organization to better collaborate with suppliers [27]. In addition, the literature review [9] revealed that it is essential to deliver high-quality software in time and within estimated cost and effort. Time-to-market is an increasingly important success factor for companies. It is important to know, which parts of the software development process are able to become more agile and how to apply the agile practices to speed up the development [16]. Atherton and Collins [28] describe challenges of product lines and compare a planned development strategy against a haphazard reuse strategy for software components. They describe the necessity of the SPL is to satisfy a wide range of customer needs [28].

Different agile methods are described to combine agile software development and a reuse strategy of software components. Well-known agile methods such as Scrum and XP are described in the literature. Furthermore, auxiliary processes are described in order to define the scope and to adjust delivery time in advance. The literature revealed that it is necessary to retest and certificate software continuously [28]. An automation of test procedures is inevitable. Continuous Integration and Continuous Testing are necessary to test new working products in each iteration in a flexible and rapid way. However, the testing tools should be capable to test each variant of software. A dynamic test strategy must be implemented.

Furthermore, the study revealed that specific crosscutting aspects such as architecture, scoping, or communication must be considered. Typically, agile development disregards a detailed consideration of the architecture. However, in product line development, architecture is an important aspect to consider [6]. It is necessary to find the right trade-off between the management of architecture evolution and refactoring without sacrificing the principles of agility. Furthermore, architecture erosion should be avoided at any time [4]. Intra- and inter-organizational communication practices and the awareness for software reuse is necessary within development teams, between teams, and the units these depend on, e.g., validation and business units [29]. It is important to keep the communication overhead low [22]. The literature review [9] confirms that in the embedded domain, safety aspects are a prime goal to satisfy [30]. Therefore, the use of agile practices in regulated and safety-critical domains is still limited [30]. Regarding regulations and standards, process descriptions and development documentation are required. A production of waste such as functionality that potentially never gets shipped to a customer should be avoided [22].

Eklund et al. [31] point out the necessity to consider different development cycle-times for hardware and mechanics. It is important to synchronize these cycles and to freeze the design at quality gates and milestones. Traditionally, no or minimal changes are allowed. Subsequent processes like optimizing the manufacturing and sales are often seen as more important. Nevertheless, software strongly depends on mechanical structures. They mention a long feedback loop with customers and management. Furthermore, a long-term predictability is hard to achieve with short-term agility [31].

Overall, we extracted 33 challenges based on the presented results in the literature review [9] (cf. Table 2).

TABLE 2. EXEMPLARY PRESENTATION OF CHALLENGES FROM THE LITERATURE REVIEW.

ID	Category	Text
136246	Challenge	It is difficult to determine, which parts of the software development process are able to become more agile.
136251	Challenge	Different variants might address different customer needs that must be satisfied.
136255	Challenge	It is a substantial effort to set up tools and to maintain the test environment.
136257	Challenge	It is necessary to find the right trade-off between the management of architecture evolution and refactoring without sacrificing the principles of agility.

4.2. Research Question 2

With the challenges entered in PTC Integrity, we were able to derive recommendations for a combination model to combine ASD and existing SPLs. We used the tracing functionality in PTC Integrity to link the recommendations. Table 3 and Table 4 are directly exported from PTC Integrity, using a customized Integrity Gateway Export Wizard. Traces from recommendations to other recommendations are shown in the columns “Related ↑” and “Related ↓”. The column “Related ↑” identifies links to the recommendations which are either a child or a refinement of the superordinate recommendation. In column “Related ↓”, the subordinate recommendations are linked. Four different traces are identified and used (cf. Figure 1). Less relevant IDs are skipped in the tables of this publication and marked as “[...]”. To present the dependencies between recommendations and challenges, even obvious main categories, such as *ID 135581 “The software shall always be tested”* are included in Table 3 to maintain the traceability.

The derived recommendations are primarily based on the challenges from the interview study and the literature review and in a second step are extended to complement the model for the combination of ASD and SPL.

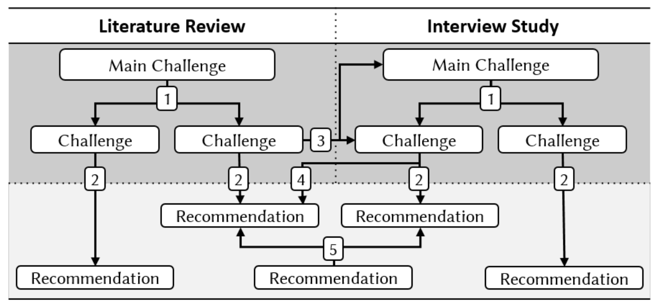

Figure 1. Traceability between Challenges and Recommendations.

Trace 1. All identified challenges were entered into a requirements document in PTC Integrity. Within PTC Integrity, we identified four different main challenges. Several particularized challenges are subsumed under the main challenges. Individual challenges are linked to the main challenge. For instance, the challenge ID136251 of satisfying the user needs is connected to the challenge ID136277 of avoiding the implementation of functionality that potentially never gets shipped to a customer (cf. Table 3). The identified challenges based on the results from the interview study [8] were grouped accordingly.

Trace 2. From the 33 challenges derived from the results of the literature review [9], 15 unique recommendations could be deduced directly. They are tagged as recommendations resulting from literature by the attribute “Source of Rec.”. The categorization is done for consistent traceability from challenge to recommendation. As shown in Table 3, the recommendation with ID166373 is based on the challenge with ID136277, which itself belongs to the main challenge ID136251. The recommendation focuses on the customer orientation of the software development. The main challange identifies the necessity to satisfy the customer and the necessity to implement the right features. This leads to the recommendation (ID136373) that each software release shall address a customer need and that the customer shall be integrated into the development process (ID135371). Another challenge (ID135873) identifies that the compliance

TABLE 3. FROM CHALLENGES TO RECOMMENDATIONS.

ID	Type	Text	Related ↑	Related ↓	Source
136251	Challenge	Different variants might address different customer needs that must be satisfied.		136277, 135371, [...]	[16]
136277	Challenge	A production of waste such as functionality that potentially never gets shipped to a customer.	136251	136373	[5]
136373	Recommendation	Each software release shall address a customer need.	136277		Literature
135371	Recommendation	The development process shall provide the customer the ability to give feedback.	136251	[...]	Literature
135873	Challenge	Maintaining the compliance to standards is seen as highly challenging.		136178, 135581, [...]	[22], Interview
136178	Challenge	Test coverage of a lot of different software variants is hard to keep high.	135873		[32]
135581	Recommendation	The software shall always be tested.	135873	135676, [...]	Interview
135676	Recommendation	A test strategy shall be implemented.	135581	135587, [...]	Interview
135587	Recommendation	Test strategy shall be capable to handle different software variants.	135576		Interview

to standards is highly challenging. This challenges is based on the findings from Kircher and Hofmann [22] and is addressed by recommendation ID135587.

Based on the results from the interview study [8], 59 individual challenges were identified. These challenges led to 54 recommendations for the combination of ASD and SPL. Most recommendations foster the introduction of agile elements into existing SPLs.

Trace 3. This trace is for literature challenges without a separate recommendation because the challenge results from the outcome of the interview study [9]. As an example challenge ID136178 from the interview study is presented. This ID verifies a challenge identified by Kowal et al. [32] for the test coverage of different software variants. The challenge identified in the literature could therefore be verified in real-life. We identified 6 challenges from the literature, which have already been addressed in the challenges from the interview study.

Trace 4. Overall, we identified 92 challenges for the combination of ASD and SPLs. 33 of them could be derived from the literature review [9] and 59 from the interview study [8]. By combining the challenges, we identified 6 from the literature, which are already addressed in the challenges from the interview study. This leads to a total of 86 individual challenges and 69 individual recommendations.

Trace 5. We introduced the category "Additional Recommendation" and considered 97 further recommendations, to complement the model for the combination. The specified recommendations are based on our experience from daily work and were reviewed by a development team from the automotive domain.

In Table 4, two examples are presented on how the "Additional Recommendation" establishes an interconnection between the individual recommendation derived from literature and the interview study.

The first example covers the recommendations which are related to synchronization processes. The collaboration with suppliers and in-house development shall be synchronized to verify a smooth agile development. From the interview study, we derived the recommendation that the organization shall provide the development team the ability to synchronize the development process (ID135323). Synchronization meetings are implemented to synchronize the development process for the in-house development (ID135460). Furthermore, software release cycles shall be synchronized (ID135669). To combine the identified recommendation, new recommendations tagged as "Additional Recommendation" are added. A synchronization meeting shall be established (ID136521) and all stakeholders shall attend the synchronization meeting. This means that according to the context the participants of the synchronization meeting (ID135460) consist of suppliers (ID135351), the development team, the purchase department, managers, and other involved departments. The synchronization meeting shall take place every month (ID135315) to verify a sprint length for development of one month. If possible, the meeting shall take place at one location. During the synchronization meeting, release planning is prepared. The software release cycles could therefore be synchronized with supplier development (ID135444) and in-house development (ID135446).

Second, we identified that Continuous Integration is the backbone of the testing process (ID135427). This is the consequence from the recommendation with ID135581. This includes and leads to further recommendation for test automation and the use of automated build servers (ID135429). A test strategy for continuous tests is essential (ID135431). With this strategy, bugs in the software are identified as early as possible. Continuous Integration with subsequent testing is essential to combine ASD and SPLs (ID135427).

5. Discussion

We approached the issue of ASD in combination with SPLs in the automotive domain and derived recommendati-

TABLE 4. INTRODUCTION OF THE CATEGORY "ADDITIONAL RECOMMENDATION".

ID	Type	Text	Related ↑	Related ↓	Source
135323	Recommendation	The organization shall provide the development team the ability to synchronize the development process.	[...]	135669, 136521	Interview
135669	Recommendation	Software release cycles shall be synchronized.	135323	135444, 135446, 140552, 140554, [...]	Literature
135444	Recommendation	Software release cycles shall be synchronized with supplier development.	135669		Additional Recommendation
135446	Recommendation	Software release cycles shall be synchronized with in-house development.	135669		Additional Recommendation
140552	Recommendation	Software release cycles shall be synchronized with the maturity of the overall vehicle, such as A, B, C, D-Samples	135669		Additional Recommendation
140554	Recommendation	Software release cycles shall be synchronized with automotive specific tests, such as summer and winter test.	135669		Additional Recommendation
136521	Recommendation	A synchronization meeting shall be established.	135323	135315, 135460, [...]	Additional Recommendation
135315	Recommendation	The synchronization meetings shall be held in a regular time interval.	136521	[...]	Additional Recommendation
135460	Recommendation	Synchronization meetings provide suppliers the ability to synchronize the development with the in-house development.	136521	135351, [...]	Interview
135351	Recommendation	Suppliers shall participate in the synchronization meetings.	135460		Additional Recommendation
135581	Recommendation	The software shall always be tested.	[...]	135427	Interview
135427	Recommendation	Continuous Integration (CI) shall be the backbone of the testing process.	135581	135429, 135413, [...]	Additional Recommendation
135429	Recommendation	Test shall be automated.	135427		Additional Recommendation
135431	Recommendation	Test shall be continuously executed.	135427		Additional Recommendation

ons for the combination. We identified 33 challenges in literature [9] and 59 resulting from an interview study [8]. Some of them from the literature review could also be identified in the interview study. Overall, we identified 86 individual challenges for the combination of ASD and SPLs within the automotive domain. For the automotive specific category, we identified 28 individual challenges, which cover topics such as the compliance to automotive standards or the deep integration between hardware and software, and safety-critical functionality. From the 86 individual challenges, we derived 69 recommendations for a combination model of ASD and SPLs. Each one of the 69 recommendations was tagged to be able to retrace to the origin of the recommendation in order to maintain the traceability of recommendations. To interconnect the recommendations based on the literature and the interview study 97 further recommendations are defined for the combination model. Those recommendations are based on working experience and best practice from the automotive domain and were reviewed within the development team. These recommendations can be used as a guideline to mitigate the risks for a combination of ASD and SPLs.

Overall, 166 recommendations could be defined. We expect that the list of recommendations is comprehensive but not complete. We expect more recommendations from the automotive domain and from regulations and standards.

6. Conclusion

This publication presents challenges for the automotive domain and furthermore described the consequences resulting from them. Additionally, challenges identified in a literature review and challenges from an interview study are compared. Recommendations are introduced to overcome the identified challenges.

A major finding of the study is that the use of agile elements in combination with software product lines requires a precise analysis of dependencies to surrounding processes and organizational structures. Dependencies between departments and suppliers are challenging and must be taken into account and well planned. Furthermore, it is important to consider processes and tasks that are necessary to meet legal requirements. Particularly, with regard to the verification and certification of software components, it is necessary to reuse software parts in the automotive sector, as legal requirements require long-term certifications.

One benefit from the recommendations is that the advantages of the software product line are not replaced by a more agile development, but extended. The identified challenges and the recommendations are the basis for an automotive-specific assessment model, addressing the combination of ASD and SPLs.

For future work, we plan to evaluate the assessment model, called the Agile Software Product Line Automotive - Model (ASPLA-Model), for the adoption of ASD in the context of existing automotive SPL development. The ASPLA-Model comprises the presented recommendations and the "Automotive SPICE Process Assessment / Reference Model" [33]. Furthermore, it focuses on the combination of ASD and SPLs and helps to integrate ASD in existing SPLs by identifying improvement potential in the current software development. It addresses all identified challenges and clusters them into seven areas to assess. According to the ASPLA assessment result, the presented recommendations are given to improve each area with respect to ASD and SPLs.

References

[1] K. Münzel, W. Boon, K. Frenken, and T. Vaskelainen, "Carsharing business models in germany: Characteristics, success and future prospects," *Information Systems and e-Business Management*, vol. 22, p. 493, 2017.

[2] M. Broy, "Challenges in automotive software engineering," in *Proc. of ICSE'06*, 2006, pp. 33–42. [Online]. Available: http://doi.acm.org/10.1145/1134285.1134292

[3] J. Bosch and P. M. Bosch-Sijtsema, "Introducing agile customer-centered development in a legacy software product line," *Software: Practice and Experience*, vol. 41, no. 8, pp. 871–882, 2011.

[4] B. Katumba and E. Knauss, "Agile development in automotive software development: Challenges and opportunities," in *Proc. of PROFES'14*, 2014, vol. 8892, pp. 33–47.

[5] M. A. Babar, T. Ihme, and M. Pikkarainen, "An industrial case of exploiting product line architecturesin agile software development," in *Proc. of SPLC'09*, 2009, pp. 171–179.

[6] J. M. Robarts, "Practical considerations for distributed agile projects" in *Conference Agile, 2008*, 2008, pp. 327–332.

[7] S. Thiel, M. A. Babar, G. Botterweck, and L. O'Brien, "Software product lines in automotive systems engineering," *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, vol. 1, no. 1, pp. 531–543, 2009.

[8] P. Hohl, J. Münch, K. Schneider, and M. Stupperich, "Forces that prevent agile adoption in the automotive domain," in *Proc. of PROFES'16*, 2016, vol. 10027, pp. 468–476.

[9] P. Hohl, J. Ghofrani, J. Münch, M. Stupperich, and K. Schneider, "Searching for common ground: Existing literature on automotive agile software product lines," in *Proc. of ICSSP'17*, 2017, pp. 70–79.

[10] U. Eklund and J. Bosch, "Applying agile development in mass-produced embedded systems," in *Agile Processes in Software Engineering and Extreme Programming*, 2012, vol. 111, pp. 31–46.

[11] J. Lantz, "Using models to scale agile mechatronics development in cars," in *Proc. of SPLC'14* , 2014, p. 20.

[12] I. F. Da Silva, P. A. Da Mota Silveira Neto, P. O'Leary, E. S. de Almeida, and S. R. De Lemos Meira, "Agile software product lines: A systematic mapping study," *Software: Practice and Experience*, vol. 41, no. 8, pp. 899–920, 2011.

[13] J. Díaz, J. Pérez, P. P. Alarcón, and J. Garbajosa, "Agile product line engineering-a systematic literature review," *Software: Practice and Experience*, vol. 41, no. 8, pp. 921–941, 2011.

[14] H. Holmström Olsson, J. Bosch, and H. Alahyari, "Towards r&d as innovation experiment systems: A framework for moving beyond agile software development," in *Proc. of IASTED multiconferences*, 2013.

[15] H. Gustavsson and U. Eklund, "Architecting automotive product lines: Industrial practice," in *Software Product Lines: Going Beyond*, ser. Lecture Notes in Computer Science, 2010, vol. 6287, pp. 92–105.

[16] M. Lindvall, D. Muthig, A. Dagnino, C. Wallin, M. Stupperich, D. Kiefer, J. May, and T. Kahkonen, "Agile software development in large organizations," *Computer*, vol. 37, no. 12, pp. 26–34, 2004.

[17] A. J. Santos and V. J. Lucena, "Scrumpl - software product line engineering with scrum," in *Proc. of ENASE'10*, 2010, pp. 239–244.

[18] J. Díaz, J. Pérez, A. Yagüe, and J. Garbajosa, *Tailoring the Scrum Development Process to Address Agile Product Line Engineering*, E.U. de Informtica (UPM), 2011.

[19] B. Fitzgerald, K.-J. Stol, R. O'Sullivan, and D. O'Brien, "Scaling agile methods to regulated environments: An industry case study," in *Proc. of ICSE'13*, 2013, pp. 863–872.

[20] A. F. Arbain, I. Ghani, and S. R. Jeong, "A systematic literature review on secure software development using feature driven development (fdd) agile model," *Journal of Korean Society for Internet Information*, vol. 15, no. 1, pp. 13–27, 2014.

[21] M. Raatikainen, K. Rautiainen, V. Myllärniemi, and T. Männistö, "Integrating product family modeling with development management in agile methods," in *Proc. of the 1st international workshop on Software development governance*, 2008, p. 17.

[22] M. Kircher and P. Hofman, "Combining systematic reuse with agile development," in *Proc. of SPLC'12*, 2012, p. 215.

[23] G. S. Neves and P. Vilain, "Reactive variability realization with test driven development and refactoring engineering, hyatt regency, vancouver, bc, canada, july 1-3, 2013," in *Proc. of SEKE'14*, 2014, pp. 100–105.

[24] M. J. Atherton and S. T. Collins, "9.2.1 developing product lines in engine control systems: Systems engineering challenges," *Proc. of INCOSE'13*, vol. 23, no. 1, pp. 184–198, 2013.

[25] J. Humble, C. Read, and D. North, "The deployment production line," in *Agile Conference, 2006*, 2006, pp. 113–118.

[26] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proc. of EASE'08*, 2008, pp. 68–77.

[27] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proc. of EASE'14*, 2014, pp. 1–10.

[28] K.-J. Stol, P. Ralph, and B. Fitzgerald, "Grounded theory in software engineering research," in *Proc. of ICSE'16*, 2016, pp. 120–131.

[29] P. Pohjalainen, "Bottom-up modeling for a software product line: An experience report on agile modeling of governmental mobile networks," in *Proc. of SPLC'11*, 2011, pp. 323–332.

[30] G. K. Hanssen, "Agile software product line engineering: Enabling factors," *Software: Practice and Experience*, vol. 41, no. 8, pp. 883–897, 2011.

[31] A. Martini, L. Pareto, and J. Bosch, "Enablers and inhibitors for speed with reuse," in *Proc.of SPLC'12*, 2012, p. 116.

[32] C. R. Jakobsen and K. A. Johnson, "Mature agile with a twist of cmmi," in *Conference Agile, 2008*, 2008, pp. 212–217.

[33] U. Eklund, H. Holmström Olsson, and N. J. Strøm, "Industrial challenges of scaling agile in mass-produced embedded systems," in *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation*, 2014, vol. 199, pp. 30–42.

[34] M. Kowal, S. Schulze, and I. Schaefer, "Towards efficient spl testing by variant reduction," in *Proc. of the 4th international workshop on Variability & composition*, 2013, p. 1.