Searching for Common Ground: Existing Literature on Automotive Agile Software Product Lines

Philipp Hohl
Daimler AG, Research and Development, Germany
philipp.hohl@daimler.com

Javad Ghofrani
Leibniz Universität Hannover, Germany
javad.ghofrani@inf.uni-hannover.de

Jürgen Münch
Reutlingen University, Germany and University of Helsinki, Finland
juergen.muench@reutlingen-university.de

Michael Stupperich
Daimler AG, Research and Development, Germany
michael.stupperich@daimler.com

Kurt Schneider
Leibniz Universität Hannover, Germany
kurt.schneider@inf.uni-hannover.de

ABSTRACT

Context: The digital transformation of the automotive industry has a significant impact on how development processes need to be organized in future. Dynamic market and technological environments require capabilities to react on changes and to learn fast. Agile methods are a promising approach to address these needs but they are not tailored to the specific characteristics of the automotive domain like product line development. Although, there have been efforts to apply agile methods in the automotive domain for many years, significant and widespread adoptions have not yet taken place. **Goal:** The goal of this literature review is to gain an overview and a better understanding of agile methods for embedded software development in the automotive domain, especially with respect to product line development. **Method:** A mapping study was conducted to analyze the relation between agile software development, embedded software development in the automotive domain and software product line development. Three research questions were defined and 68 papers were evaluated. **Results:** The study shows that agile and product line development approaches tailored for the automotive domain are not yet fully explored in the literature. Especially, literature on the combination of agile and product line development is rare. Most of the examined combinations are customizations of generic approaches or approaches stemming from other domains. **Conclusion:** Although, only few approaches for combining agile and software product line development in the automotive domain were found, these findings were valuable for identifying research gaps and provide insights into how existing approaches can be combined, extended and tailored to suit the characteristics of the automotive domain.

CCS CONCEPTS

•**Software and its engineering → Software product lines;**

KEYWORDS

software product line, agile methods, automotive, literature review

ACM Reference format:
Philipp Hohl, Javad Ghofrani, Jürgen Münch, Michael Stupperich, and Kurt Schneider. 2017. Searching for Common Ground: Existing Literature on Automotive Agile Software Product Lines. In *Proceedings of International Conference on Software and System Processes, Paris, France, July 2017 (ICSSP'17),* 10 pages.
DOI: 10.475/123_4

1 INTRODUCTION

Agile development within Software Product Lines (SPLs) is not new. Several studies treated this issue before. Perez and Silva [15] conducted a mapping study and a structured literature review on agile software product line engineering. Our study extends the findings from the survey and includes references published after 2011. We primarily focus on literature which addresses the automotive domain. The aim of our study is to increase the understanding on how agile software development (ASD) might be integrated within an automotive software product line. We limited the scope to automotive embedded software development. In this context it is important to consider deep integration between hardware and software, strong focus development processes, strong supplier involvement, and safety-critical functionality [17]. Furthermore, specific testing conditions, like tests in the real car, must be considered.

The presented study examines existing challenges which arise when agile development and SPLs are introduced in the automotive domain and how they can be overcome.

Existing literature deals with (1) the use of agile methods in the automotive domain, (2) the use of product line development in the automotive domain, and (3) the combination of agile methods and product line development. This study offers an intersection of all three topics. Furthermore, the literature review helps to identify already existing solutions in published literature.

The remainder of this paper is structured as follows. Section 2 describes the method used for the literature review. Section 3 reports the contributions of the collected studies. Section 4 summarizes key findings as well as implications for practitioners. Section 5 presents conclusions due to the findings. Finally, Section 6 summarizes the paper and gives an outlook on future work.

ICSSP'17, Paris, France
© 2017 ACM. 123-4567-24-567/08/06…$15.00
DOI: 10.475/123_4

2 RESEARCH DESIGN

In order to conduct a literature review, two main research methods are applicable: A mapping study as well as a structured literature review. Both methods share some commonalities. For the presented study, the researchers conducted a mapping study as this method enables to identify gaps in literature to the specific research field [16, 56]. In addition, the mapping study allows to gain a wide overview of the research area [36]. The research process described by Petersen et al. [56] served as a basis for our research. Researcher 1 (P. Hohl) was following the process primarily. This process was extended by the data collection of Researcher 2 (J. Ghofrani). He conducted a snowballing search to find related literature which helps to answer the research questions. By including this literature and combining it with the results from Researcher 1, we achieved a broader view on the topic. With this, related research areas are taken into account. Whenever we deviated from the process, we describe our actions in the following subsections.

The process by Peterson et al. [56] is illustrated in Figure 1, whereas solid lines and white background define *Process Steps* and dashed lines with grey background the corresponding *Outcome* to each process step.

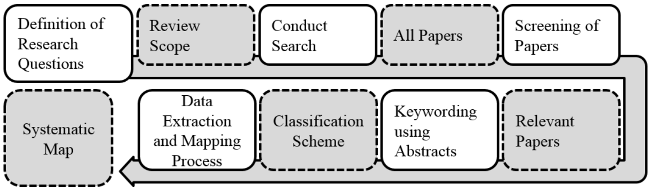


Figure 1: Research process by Petersen et al. [56]

In the first step of the research process, research questions were defined. In the next step, we conducted the search within suitable reference databases. Therefore, a search string was constructed. We extended the results by two different searches, conducted by two researchers independently. In the third step, the papers were screened by applying defined inclusion and exclusion criteria. Whenever, it was not clear how to categorize a paper, it was discussed by Researcher 1 and Researcher 2 in a review meeting. The abstracts were screened in detail and a mapping process was executed as a last step in the mapping process.

2.1 Research questions

The general question that guides this mapping study is: *What is the state-of-the-art to combine agile software development and software product lines in the automotive domain, according to published literature?* The question is divided into three research sub-questions to provide different views on the topic. Figure 2 summarizes the questions.

RQ 1: *In what ways can software product lines be combined with existing agile software development in the automotive domain, according to published literature?* The goal is to investigate how ASD in the automotive domain could be extended by a SPL. Ideas, found in the existing literature, are evaluated, which foster an integration and focus on existing challenges for the introduction of a SPL.

RQ 2: *In what ways can agile software development be combined with existing software product lines in the automotive domain, according to published literature?* This question investigates how current SPL development in the automotive domain could be extended by agile development approaches. Ideas found in the existing literature are evaluated, which foster an integration and focus on existing challenges to introduce ASD.

RQ 3: *Are there any suitable concepts from agile software product lines to adapt within the automotive domain, according to published literature?* To complete the picture, this question identifies how concepts for Agile Software Product Lines (ASPLs) and agile embedded software development can be adapted to the automotive domain.

2.2 Conducting the search

This section describes the research process for the mapping study executed by Researcher 1 and, furthermore, describes the research process applied by Researcher 2. The individual results were combined and screened as described in Section 2.3.

2.2.1 Search process applied by Researcher 1. The used research process based on Petersen et al. [56] is divided into five steps. These steps were executed in a sequential order:

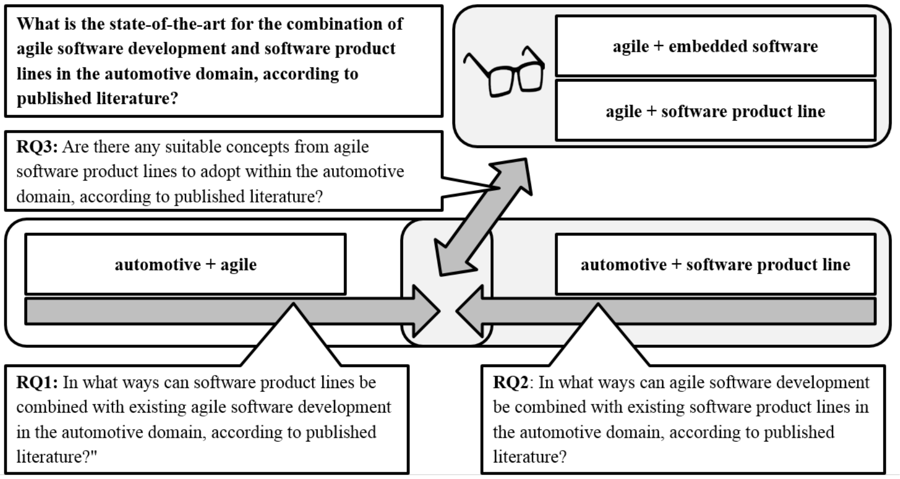
Selection of databases. The selection of databases is essential to make the research outcome as comprehensive as possible. Eight databases were selected for the mapping study: Scopus, Science Direct, IEEE, ACM DL, Springer Link, Wiley Online Library, World Scientific, and Google Scholar. With these databases, a comprehensive search was conducted to avoid bias [56]. Although Scopus covers IEEE Xplore and Elsevier, the two databases were included to verify the quality of the search strings.

Definition of the search string. The construction of the search string for the mapping study was conducted as follows: First, three main keywords were selected. For these three keywords, synonyms and a corresponding word family were defined (cf. Table 1).

Table 1: Overview search keywords

Search keyword 1	”software product line(s)”
Synonyms	”software product line engineering”
	”product line(s)”
	”software product line development”
Word family	”software product line engineering”
Search keyword 2	”automotive”
Synonyms	”regulated domain(s)”
Word family	”car manufacturing”
	”car development”
Search keyword 3	”agile software development”
Synonyms	”agile methods”
	”agile practices”
Word family	”agile software development”

The constructed search string: *(”software product line” OR ”software product lines” OR ”software product line engineering” OR fisoftware product line development” OR ”product line” OR ”product lines”) AND (”automotive” OR ”regulated domain” OR ”regulated domains”*


What is the state-of-the-art for the combination of agile software development and software product lines in the automotive domain, according to published literature?

RQ3: Are there any suitable concepts from agile software product lines to adopt within the automotive domain, according to published literature?

RQ1: In what ways can software product lines be combined with existing agile software development in the automotive domain, according to published literature?"

RQ2: In what ways can agile software development be combined with existing software product lines in the automotive domain, according to published literature?

agile + embedded software
agile + software product line
automotive + agile
automotive + software product line

Figure 2: Directions of research questions

OR ”car manufacturing” OR ”car development”) AND (”agile software development” OR ”agile methods” OR ”agile practices”)

The shape of the search string was adapted to the selected databases, since search options differ and are specific for each search engine. The search was conducted using titles, abstracts and keywords.

2.2.2 Search process applied by Researcher 2. Researcher 2 conducted a systematic literature review using the snowballing method as a systematic approach. Therefore, he was following the ”Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering” by Wohlin [72].

Identify a start set of papers. This step identifies a start set of papers in order to use it for the snowballing procedure. The start set consists of literature which is related to the research. Wohlin [72] mentioned, that there exists no silver bullet for identifying the start set. To conduct the search it is necessary to include relevant papers. Those papers may come from different communities.

Selection of databases. Selecting databases which contain the related scientific studies for ASD methods and software product line engineering is an important factor to achieve valid search results. Five databases were selected for the creation of the start set: ACM Digital Library, Web of Science, IEEE Xplore, Science Direct, and Wiley Inter Science.

Building a start set. An initial search string is built from the combination of related keywords to software product line development and agile software development. Keywords and corresponding words are defined in Table 2. This results in the following search string: *(”Agile” OR ”SCRUM” OR ”Kanban”) AND (”software”) AND (”product line” OR ”product family” OR ”reuse” OR ”variability”) AND (”methode” OR ”development” OR ”progress” OR ”engineering”)*

The shape of the search string was adapted to the selected databases. In total, 26 studies were identified and build up the start set for the Systematic Literature Review using the snowballing approach.

Table 2: Overview search keywords for building the start set

Search keyword 1	”Agile”
Related keywords	”SCRUM”
	”Kanban”
Search keyword 2	”software”
Related keywords	no related keyword selected
Search keyword 3	”product line”
Related keywords	”product family”
	”feature”
	”reuse”
	”variability”
Search keyword 4	”methode”
Related keywords	”development”
	”progress”
	”engineering”

Performing the snowballing. For snowballing, the ReviewR tool, was used. This tool is an in-house development of the Software Engineering Group of Leibniz Universität Hannover to conduct snowballing. The tool manages the search iterations and performs the snowballing within a Scopus database. Five iterations with forward and backward snowballing were performed. If the title and the abstract of the publication contains any combination of agile methods and SPLs it is included for the next iteration. We excluded duplicates, studies older than 2001 and studies which are only mention one keyword.

2.2.3 Definition of inclusion and exclusion criteria. To eliminate studies which are not relevant for answering the research questions, exclusion criteria were applied [56].

Inclusion criteria:

- The paper discusses agile software product line development in the automotive domain.
- The paper discusses product line engineering in the automotive domain.
- The paper discusses techniques to speed up feature development in the automotive software development.

- The paper discusses agile software development for embedded software.
- The paper discusses agile software development in combination with the use of SPLs.
- The paper is written in English or German.

Exclusion criteria:

- The paper was written before 2001.
- The paper has no accordance with at least two of the search keywords.
- The paper focuses on a specific tool. (This study refers to methods and not to specific tools.)
- The paper is not accessible.
- Duplicated papers.

2.2.4 Selection of researchers to conduct the study. Due to the fact that the mapping study is part of a PhD project, Researcher 1 conducted the mapping study. Researcher 2 conducted a separate study and added his literature to extend the dataset and to give a broader view with respect to the related research areas.

2.2.5 Execute the research. As described in the previous steps, the research was executed in the selected databases.

2.3 Screening of papers

In this step, the previously defined inclusion and exclusion criteria were used to select the most suitable set of papers. The screening process was divided into several filtering steps as shown in Figure 3. Papers were included if they fulfilled the inclusion criteria. This filtering was applied to the results of Researcher 1 and Researcher 2.

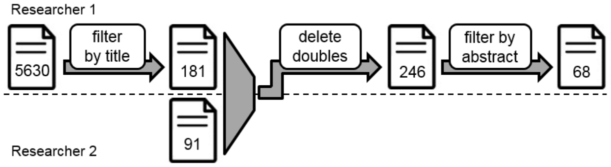


Figure 3: Filtering steps

The search of Researcher 1 resulted in 5630 identified contributions. In the first step of the screening process, the inclusion and exclusion criteria were applied on the title and keywords of the identified studies. Duplicate papers were excluded as well. Whenever Researcher 1 was not able to include or exclude the paper, he marked the paper as "unclear". In this filtering step, 23 papers were classified as unclear. These papers were reviewed by Researcher 2 and categorization was discussed with Researcher 1 in a review meeting. If the paper was still unclear after this process, another researcher reviewed the paper and decided to include it to one of the given categories or exclude it. This filtering step leads to 181 remaining papers.

In addition, Researcher 2 separately searched with a broader approach (cf. Section 2.2.2) to find successful implementations and experience reports examining the combination of ASD and the use of SPLs. This search resulted in a total number of 91 resources.

Both data sets were combined and Researcher 1 deleted the double references in the dataset. For the remaining 246 references, the abstracts were extracted and stored in the reference management and citation tool Citavi. Researcher 1 evaluated the abstracts. This last filtering step led to a final set of 68 papers.

2.4 Categorizing the studies

For later data extraction, the results of this study were categorized. To analyze, categorize and to sort the collected literature, we used the tagging and coding functionality in the reference management and citation tool Citavi. The first step in Citavi was to read abstracts and identify keywords and concepts reflecting the contribution of the papers [56]. The selected keywords were used for clustering the papers into categories [56]. According to Figure 2, three main categories were defined. The output of this step gave an overview of the covered topics and areas of the selected papers and defined the current state of research at a granular view.

2.5 Data extraction

For each specified category, the data of each paper was extracted. The first step of data extraction gathered standardized information like author, type of paper and publication channel. This information was used to give an overview about the active participants in the considered area of research. Furthermore, the results showed frequencies of papers for each category. This enables us to identify gaps and possibilities for future research [56].

After getting an overview of the papers, detailed data was extracted in order to answer the research questions.

2.6 Threats to validity

This section treats the identified threats to the validity of our study.

Research questions: We defined the research questions to be unambiguously understandable. Therefore, we reviewed the wording of the research questions several times. It would be false to affirm that our research questions cover the complete existing research on ASPLs, containing all related research areas. However, our research questions address an urgent issue in that field.

Publication bias: Due to the fact, that only Researcher 1 was doing the screening process, we can not guarantee that all relevant primary studies were selected. It might be possible that papers were excluded in the filtering steps. For the data collection, we mitigated this by adding the literature from Researcher 2. Furthermore, a review session was introduced, to discuss the "unclear" marked publications.

Search conducted: The shape of the search string was adapted to the selected databases, since search options differ and are specific for each search engine. We mitigate the threats introduced by this adaption by using eight databases. Although the database Scopus covers IEEE Xplore and Elsevier, the two databases were included to verify the quality of the search strings. This threat was mitigated further, by adding the literature from Researcher 2.

Data extraction: Data extraction and categorization of the papers was done by Researcher 1. This could introduce bias and incorrectly classified papers. Researcher 1 conduct this classification two times to avoid a wrong categorization.

3 REPORTING THE REVIEW

To visualize the results, we divided them into the three categories from Figure 2. The following Table 3 contains the number of studies according to the publishing date. As shown in Table 3, a lot of research focus on related research areas. The number of papers have increased since 2004. It furthermore illustrates that papers are rare with respect to the automotive domain. Only in 2014, four papers focus on ASD and address the automotive domain.

Table 3: Publications by year

Year	Automotive & Agile	Automotive & SPL	Related Research Area
2004			[40, 41]
2005			
2006			[37, 55, 66, 68]
2007			[31, 50]
2008	[70]		[23, 26, 44, 45, 49, 58, 67]
2009		[64]	[4, 6, 25, 42, 51]
2010			[22, 24, 47, 59]
2011	[39]		[5, 7–9, 11, 14, 15, 28, 38, 57]
2012		[30]	[10, 17, 32, 35, 43, 52, 61, 69]
2013		[18]	[3, 33, 53]
2014	[1, 34, 62, 63]	[13]	[2, 19–21, 27, 46, 48, 65]
2015	[29, 71]	[73]	[12]
2016	[60]		

However it seems, that no paper handles the combination of ASD in the automotive domain with respect to SPL development. This will be analyzed in more detail. The following subsections will derive results from the literature review to answer the research questions:

3.1 Research Question 1

In what ways can software product lines be combined with existing agile software development in the automotive domain, according to published literature?

The published literature does not provide significant information or approaches on how to adapt existing ASD approaches to SPL development in the automotive domain. Therefore, it is difficult to answer the research question based on information from existing literature. However, several publications were found that focus on the adaptation of ASD approaches to the specifics of the automotive domain. As SPLs are a common approach in the automotive domain it could be argued that this adaptation might at least implicitly consider also product line development. The findings can be summarized in the following: (1) Adapting ASD in the automotive typically concentrates only on selected agile elements such as Continuous Integration (CI). The published literature does not show any recommendations to use a comprehensive set of agile elements and practices together in the automotive domain. (2) The majority of the published literature suggests that agile models and processes should get customized to the specifics of the automotive domain before they are implemented in practice. (3) Several agile models and processes that are already customized to the specifics of the automotive domain are proposed in the published literature. Examples are the Feedback Loop Model that especially considers the collaboration between different organizations (such as OEMs and suppliers) and the Agile Hybrid Assessment Method that especially considers safety-criticality constraints in the automotive industry. These methods and processes include interesting new concepts such as virtual integration on the system level.

Thiel et al. [64] analyze the combination of ASD and plan-driven processes which could be seen as a typical characteristic of current automotive software development. They emphasize that a combination could be beneficial under certain conditions such as rigid quality and safety requirements [64]. As a solution, Thiel et al. [64] suggest to introduce selected agile approaches to automotive systems engineering. This suggestion is also supported by the "Agile in Automotive Survey" from Kugler and Maag [71]. The survey highlights that many companies recommend to only introduce single agile elements into existing and proven development cycles. In addition, they recommend not to apply a purely SCRUM approach in practice. Instead, they emphasize that pure SCRUM is often not applicable due to the context factors and the development environment.

Schlosser et al. [60] define an upcoming challenge in the automotive software development. They identify the necessity of shorter development time for multiple software variants by considering constraints of a high cost pressure. They further argue that more incremental software deliveries are necessary for a quick response from customer which shortens the development time and save money [34, 60]. In addition, the agile element of CI provides a quick response with respect to functional aspects [1, 60, 70]. In order to integrate further agile elements into the automotive domain, different models and processes are introduced including a (1) *Feedback Loop Model*, the (2) *Agile, Hybrid Assessment Method for the Automotive Industry* and the (3) *Mega Scale Software Product Line Engineering*:

The concurrent *Feedback Loop Model* (1) introduces feedback loops for different organizations to enable cooperation [29]. The cooperation (including communication) between different organizations is managed by a new architect role to shorten the development time. The reduction of time is validated in a case study. In turn, the case study reveals several drawbacks [29].

The *Agile, Hybrid Assessment Method for the Automotive Industry (AHAA)* (2) considers safety critical development. Software quality is important for software critical companies like in the automotive domain to ensure high reliability and maturity of software. Plan-driven practices are assessed with respect to software process improvement like CMMI and ASPICE [63].

Additionally, the assessment suggests agile based improvement solutions within the automotive domain. Furthermore, the *System Architecture Virtual Integration* (SAVI) initiative helps to improve the agile development. In SAVI, CI is concurrently operated within the software development process. Integration starts with the earliest available system models into a virtual integration environment. With a virtual integration strategy vendors are more closely tied to the project [62].

3.2 Research Question 2

In what ways can agile software development be combined with existing SPLs in the automotive domain, according to published literature?

Research Question 2 could not be answered by the selected findings. The majority of the published literature is concerned with SPLs within the automotive domain without considering an adaptation to ASD. Several variant management approaches and appropriate methods and processes to handle variants are discussed. Some product line approaches are presented that might be helpful in understanding what to consider when combining with agile development. Understanding binding times, for instance, might help to better organize CI mechanisms.

According to Wozniak et al. [73], the automotive domain is the most challenging environment for systems and SPL engineering. Millions of different software variants exist, where each constraints a large complexity. The complexity is based on the large number of variation points within the product [21]. Four different binding times may be distinguished: programming, integration, assembly, and runtime. All of them are used in the automotive domain. This leads to a "very large number of individually complex products with incomparably rich feature variation among them" [73, p.336]. In order to handle the complexity, the Mega Scale Software Product Line Engineering (MS-SPLE) (3) approach is introduced. This approach is applied for large product sets with complex products and complex feature variations like the automotive domain. MS-SPLE is a possibility to manage, e.g., calibration parameters for the software variation mechanism and the complexity management. However, ASD is not taken into account within MS-SPLE [73].

3.3 Research Question 3

Are there any suitable concepts from agile software product lines to adopt within the automotive domain, according to published literature?

The published literature provides several sources of information on concepts for the combination of ASD and SPLs. These concepts could be used as candidates for the adoption in the automotive domain or they might reveal helpful insights for an adaptation in the automotive domain. The findings can be summarized as follows: (1) Several concrete methods and models are described that combine agile and product line concepts. An example is ScrumPL which supports iterative domain and application engineering. (2) Specific cross-cutting aspects such as architecture, scoping, or communication are considered in the published literature with respect to the combination of agile and product line development. An example is a scoping approach that allows conducting a commonality and variability analysis in each iteration in order to constantly understand the implications of new or changing requirements. (3) Several publications present expected or experienced benefits of agile product line development. Among others, frequently mentioned benefits are increased flexibility, shortened development cycles, and in consequence a reduction of technological, customer, and market risks. (4) Different strategies exist to introduce a combination of agile and product line development. Although it is difficult to say with certainty, the primary strategy seems to be introducing agile into well-established SPL development. (5) The combination of agile and product line development significantly depends on customer needs (such as availability or cost efficiency).

Besides concepts from agile product line development the published literature includes information on agile development in embedded domains. This information might be helpful for transitioning towards agile product line development in the automotive domain.

In 2007, Hummel and Atkinson [31] stated, that agile development and systematic software reuse have rarely been attempted in the same project. Pohjalainen [57] describes problems with the combination of agile and product line development based on the required upfront planning in SPLs, while agile method practitioners inherently want to avoid any heavy up-front planning [57]. To overcome this incompatibility, Pohjalainen [57] describes a bottom-up approach to combine product line engineering and formal modeling with ASD.

Martini [43] mentioned, that agile and reuse practices do not hinder each other. Some sources emphasize the benefits of introducing ASD practices to SPLs [26]. Potential benefits include the improved change management for requirements, the increased product quality with coincident decrease of development costs, or a reduced time to market [20]. The development enriched by agile practices is often referred to as *Agile Product Line Engineering* (APLE) [48].

Mainly two approaches for introducing APLE are reported: Either take an existing SPL process and introduce agility or take an agile process in order to tailor it for SPLs [66]. The focus is primarily on introducing agile into well-established SPL rather than on the implementation of software reuse into an agile-oriented development [43]. It is essential to develop a reuse strategy that complements the principles of agile development and to foster software reuse in combination with the practices of agile methods [31]. Hanssen [28] 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 collaborate better with suppliers [28]. The published literature can be structured by two different viewpoints: The technical viewpoint and the business viewpoint.

Regarding the business viewpoint, it is often mentioned that it is essential to deliver high-quality software in time and within estimated cost and effort as time to market is an increasingly important success factor for companies. It is, therefore, 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 [40].

One main goal with respect of the SPL is to satisfy a wide range of customer needs [3]. Atherton and Collins [3] describe challenges of product lines and compare a planned development strategy against a haphazard reuse strategy for software components in aircraft engines. They describe different needs of military customers, who focus on capability and system availability. In addition, they consider commercial customers, who focus on minimizing life cycle cost. McGregor [45] emphasizes, that an entire customer satisfaction always requires a tradeoff between flexibility and software waste due to late rework of code and a late binding time of variability [38, 46].

The technical viewpoint can be divided into *agile methods and models*, which are already in use and *agile aspects* like documentation or scoping, which should be considered for a successful combination.

3.3.1 Agile methods and models. 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, various papers present a tailored approach to use well-known agile methods. Lindvall et al. [40] describe a project which utilizes the XP approach. This is achieved by auxiliary processes in order to define the scope and to adjust delivery time in advance. With respect to Scrum, a new method called ScrumPL is introduced. Scrum lifecycle phases and the SPLE sub-processes are combined to form ScrumPL [59]. In this tailored Scrum, domain engineering as well as application engineering processes are performed in an iterative way separated in different sprints [14].

Another approach of combining SPLs and agile development is the *Feature Driven Approach* with the use of *Feature Models* [2, 58]. This hybrid method captures the initial view of the results of the commonality and variability analysis [45]. In this approach, both disciplines are merged. As a result, feature orientation blends the benefits of product line engineering with those of agility [35]. As an extension of the feature model, the composite feature model is described by Urli et al. [69]. Their solution relies on the definition of composite feature models and the use of a model-driven evolution process to support it on large real systems [69]. In addition, Trinidad et al. [68] evaluate the possible reasons why a feature model has dead features and how to evaluate feature models in general.

Neves and Vilain [48] present the idea of *Test-Driven-Development* (TDD) as the basis to combine ASD and SPLs. The usage of refactored agile practices lead to a reactive SPL. Therefore, SPL evolves and acquires variability points on demand [48]. Due to the speed up of the development, it is necessary to retest and certificate some parts repeatedly [3]. It is, therefore, essential to automate test procedures. *Continuous Integration* and *Continuous Testing* (CT) are necessary to test new working products in each iteration in a flexible and rapid way. However, in agile methodologies, testing tools should be capable to test the developed software. It is a significant effort to set up tools and to maintain the test environment [42]. Ghanam et al. [26] extend the idea of TDD and describe tests not only in form of regression tests. They rather describe the test approach as a way to determine commonalities in software systems of the same domain [26].

3.3.2 Agile aspects. This section presents cross-cutting aspects like architecture, scoping, communication and safety regulations. These aspects should be considered in detail when combining ASD and SPLs.

Architecture. McGregor [45] suggests an architecture definition practice that replaces the product line and agile practices focusing on the structure, commonality and variability. Typically, agile development disregards a detailed consideration of the architecture. However, in product line development, architecture is an important aspect to consider [4]. It is necessary to find the right tradeoff between the management of architecture evolution and refactoring without sacrificing the principles of agility. Furthermore, architecture erosion should be avoided at any time [8]. The use of proprietary architectures could help to avoid this. In contrast, Chong [9] promotes the adoption of open standards, rather than closed proprietary architectures. He mentions this as an essential economic driver for opening closed markets to competition [6, 9].

Scoping. Ghanam et al. [22] introduce a new lightweight variability analysis technique. This enables to determine commonalities and variations between the new requirements and the existing ones. The analysis is conducted in every agile iteration and not as traditional up-front [22]. In addition, another lightweight approach is presented by Atherton and Collins [3]. They divide requirements into small pieces of functionality, categorize and priories them [3].

Communication. An important key factor for a successful combination of ASD and SPLs is communication [27]. Intra- and inter-organizational communication practices and the awareness for software reuse are essential [43]. Good communication is necessary within development teams, between teams, and the units these depend on, e.g., validation and business units [43]. It is important to keep the communication overhead low [35].

Safety regulations. Kirchner and Hofman [35] conduct a study based on findings revealed in the area of medical device development. In this area, they have to fulfill certain pre-requisites with regards to patient safety. Development organizations have to adhere regulations to comply with the law [35]. Regarding regulations and standards, process descriptions and development documentation are required. However, this may become inefficient if the overhead to organize document writing, signing, reading, reviewing, and revising increase. A production of waste such as functionality that potentially never gets shipped to a customer should be avoided [35].

Regarding the combination of agile and embedded development, the published literature reveals that the motivation for using agile methods is often characterized by market uncertainties, competitive pressures, and the need to shorten development cycles [53].

In the embedded domain, safety aspects are a prime goal to satisfy [32]. Therefore, the use of agile practices in regulated and safety-critical domains is still limited [32]. With respect to the literature, the study of Kaisti et al. [33] finds no reason why agile methods could not be used in the embedded domain. XP and Scrum or a mixture of both are the wide-spread used methods [61]. Eklund et al. [19] 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 a quality gate and milestone. Traditionally, no or minimal changes are allowed. Subsequent processes like optimizing the manufacturing and sales are often seen as more important. Nevertheless, software is strongly dependent on mechanical structures [19]. They further mention a long feedback loop with customers and management. Furthermore, a long-term predictability is hard to achieve with short-term agility [19].

4 DISCUSSION

We approached the issue of ASD in combination with SPLs in the automotive domain from three different perspectives.

The first direction was treated by Research Question 1. The question focuses on existing ASD in the automotive domain. It further examines, if SPL development could be combined to already existing solutions. Although none of the identified approaches addresses the existence or adaption of SPLs, we identified several approaches in order to integrate agile elements into the automotive domain.

The second approach to gain knowledge about the combination of all three search keyword, was treated in Research Question 2. Here we focus on existing SPLs in the automotive domain and spot on the combination with ASD. The results we found describe SPLs within the automotive domain but do not consider agile development approaches. A possible explanation is that combining agile development with existing product lines is challenging (consider, for instance, the need for synchronization with mechanical and electrical engineering). This might be the reason that the advent of agile methods in the automotive industry did not start in the area of product line development and no published literature is available on this aspect. Vice versa, there are areas in the automotive domain (e.g., entertainment software for the car) where it seems to be comparably easy to introduce agile development without product line development. This might explain that the study was more fruitful with respect to Research Question 1. Anyhow, the need to get more agile in the automotive domain and the lack of approaches with respect to Research Question 2 creates a very strong motivation to do a case study or some other empirical research in order to get answers to this question.

A key finding is that there exists no ready-made solution to combine ASD with SPLs in the automotive domain. It is therefore necessary to take the related research areas into account. In RQ 3, we identified challenges and possible solutions, which could be relevant for the automotive domain as well. The identified solutions given by the literature might be transferred to the automotive domain (cf. Table 4).

5 CONCLUSION

The literature review revealed that there is no approach specifically tailored to the automotive domain handling the combination of ASD and SPLs.

Searching for existing literature about the introduction of agile methods in the context of automotive product line development was also not fruitful. Searching for literature that deals with introducing product line engineering to agile development in the automotive domain also did not reveal significant results.

However, looking at related domains provided many insights into how the combination of agile methods and product line development could look alike in the automotive domain. These insights might help as starting points to modify automotive software development processes so that they can benefit from both, agility and systematic reuse through product lines.

6 SUMMARY AND OUTLOOK

Our work provides an overview on agile and SPL approaches in the automotive domain as well as their common ground. It further identifies the current gaps in the research and proposes specific agile methods and practices for integration into automotive SPLs. Moreover, our work concludes that there is an existing gap in the literature with respect to agile SPL development in the automotive domain.

Altogether, our work reveals many fields for further research, such as organizational, process, and product-oriented challenges. It further helps to identify means for addressing specific characteristics and restrictions of the automotive domain.

For future work, we aim at transferring the presented solutions to the automotive domain and evaluate them in existing development projects. As a result, we plan to create and evaluate a model for the adaption of ASD in the context of existing automotive SPL development.

REFERENCES

[1] V. Antinyan, M. Staron, W. Meding, P. Osterstrom, E. Wikstrom, J. Wranker, A. Henriksson, and J. Hansson. 2014. *Identifying risky areas of software code in Agile/Lean software development: An industrial experience report.* In *CSMR-WCRE.* pp. 154–163. DOI: http://dx.doi.org/10.1109/CSMR-WCRE.2014.6747165

[2] A. F. Arbain, I. Ghani, and S. R. Jeong. 2014. *A Systematic Literature Review on Secure Software Development using Feature Driven Development (FDD) Agile Model.* In *Journal of Korean Society for Internet Information* Vol. 15, pp. 13–27. DOI: http://dx.doi.org/10.7472/jksii.2014.15.1.13

[3] Atherton, M. J. and Collins, S. T.. 2013. *Developing Product Lines in Engine Control Systems: Systems Engineering Challenges.* In *INCOSE Int. Symposium* Vol. 23, pp. 184–198. DOI: http://dx.doi.org/10.1002/j.2334-5837.2013.tb03012.x

[4] M. A. Babar, T. Ihme, and M. Pikkarainen. 2009. *An Industrial Case of Exploiting Product Line Architectures in Agile Software Development.* In *Proc. 13th SPLC.* pp. 171–179.

[5] M. Balbino, E. S. de Almeida, and S. R. de L. Meira. 2011. *An Agile Scoping Process for Software Product Lines & Knowledge Engineering.* pp. 717–722.

[6] S. Black, P. P. Boca, J. P. Bowen, J. Gorman, and M. Hinchey. 2009. *Formal Versus Agile: Survival of the Fittest.* In *Computer* Vol. 42, pp. 37–45. DOI: http://dx.doi.org/10.1109/MC.2009.284

[7] B. Blau and T. Hildenbrand. 2011. *Product Line Engineering in Large-Scale Lean and Agile Software Product Development Environments - Towards a Hybrid Approach to Decentral Control and Managed Reuse.* In *Proc. 6th ARES.* pp. 404–408.

[8] J. Bosch and P. M. Bosch-Sijtsema. 2011. *Introducing agile customer-centered development in a legacy software product line.* In *Software: Practice and Experience* Vol. 41, pp. 871–882. DOI: http://dx.doi.org/10.1002/spe.1063

[9] S. Chong, C.-B. Wong, H. Jia, H. Pan, P. Moore, R. Kalawsky, and J. O'Brien. 2011. *Model Driven System Engineering for vehicle system utilizing Model Driven Architecture approach and hardware-in-the-loop simulation.* In *Proc. ICMA 2011.* pp. 1451–1456.

[10] I. F. Da Silva. 2012. *An agile approach for software product lines scoping.* In *Proc. 16th SPLC.* pp. 225-228. DOI: http://dx.doi.org/10.1145/2364412.2364450

[11] I. F. Da Silva, P. A.Da Mota Silveira Neto, P. O'Leary, E. S. de Almeida, and S. R.De Lemos Meira. 2011. *Agile software product lines: A systematic mapping study.* In *Software: Practice and Experience* Vol. 41, pp. 899–920. DOI: http://dx.doi.org/10.1002/spe.1078

[12] I. F. Da Silva, P. A. Da Mota Silveira Neto, P. O'Leary, E. S. De Almeida, and S. R. De Lemos Meira. 2015. *Using a multi-method approach to understand agile software product lines.* In *Information and Software Technology* Vol. 57, pp. 527–542. DOI: http://dx.doi.org/10.1016/j.infsof.2014.06.004

[13] I. F. Da Silva, P. A. Da Mota Silveira Neto, P. O'Leary, E. S. De Almeida, and S. R. Meira. 2014. *Software product line scoping and requirements engineering in a small and medium-sized enterprise: An industrial case study.* In *Journal of Systems and Software* Vol. 88, pp. 189–206. DOI: http://dx.doi.org/10.1016/j.jss.2013.10.040

[14] J. Díaz, J. Pérez, A. Yagüe, and J. Garbajosa. 2011. *Tailoring the Scrum Development Process to Address Agile Product Line Engineering.*

[15] J. Díaz, J. Pérez, P. P. Alarcón, and J. Garbajosa. 2011. *Agile product line engineering-a systematic literature review.* In *Software: Practice and Experience* Vol. 41, pp. 921–941. DOI: http://dx.doi.org/10.1002/spe.1087

[16] A. Dresch, D. P. Lacerda, and J. A. V. Antunes Jr.. 2015. *Design Science Research.* DOI: http://dx.doi.org/10.1007/978-3-319-07374-3

[17] U. Eklund and J. Bosch. 2012. *Applying Agile Development in Mass-Produced Embedded Systems.* In *Agile Processes in SE and Extreme Programming.* pp. 31–46. DOI: http://dx.doi.org/10.1007/978-3-642-30350-0_3

[18] U. Eklund and H. Gustavsson. 2013. *Architecting automotive product lines: Industrial practice.* In *Science of Computer Programming* Vol. 78, pp. 2347–2359. DOI: http://dx.doi.org/10.1016/j.scico.2012.06.008

[19] U. Eklund, H. Holmström Olsson, and N. J. Strøm. 2014. *Industrial Challenges of Scaling Agile in Mass-Produced Embedded Systems.* In *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation.* Vol. 199. pp. 30–42. DOI:

Table 4: Challenges identified in related research areas and potential solutions for the automotive domain

Identified Challenge:	Market uncertainties and fast changing requirements.[2, 5, 6, 20–22, 37, 38, 41, 47, 48, 51, 54, 65–67]
Possible Solution:	Faster customer feedback and agile development to be more flexible in the development.
Potential Transfer:	- Set up a new feedback culture for customers. This leads to a restructuring in the organization and the possibility for the project managers to act as internal customers to promote ideas and new features. - Set up an idea exchange forum for customers to submit new ideas or improvements. -Get direct customer feedback in beta-test drives on a separate test route.
Identified Challenge:	Competitive pressure and the need to shorten development cycles. [4, 8, 20, 37, 38, 41, 47, 49, 50, 52, 54]
Possible Solution:	Innovative features implemented in an agile way could be delivered faster to foster business advantages.
Potential Transfer:	- A deployment of beta-software to the customer is hard to apply. Restrictions given by the law and relevant safety issues are of highest priority. The use of SPLs helps to reuse already certificated software parts and to save time. - Auxiliary tasks like CI help to ensure high quality software due to automated software test at a high pace.
Identified Challenge:	Hard to change the organization. [19, 40, 41]
Possible Solution:	Stepwise integration to integrate agile practices and methods within the software teams. Self-organizing teams.
Potential Transfer:	- Integrate agile practices in an incremental and stepwise way. - Introducing single agile elements into existing and proven development cycles by keeping the existing SPL to retain the software quality.
Identified Challenge:	Different development cycle-times for related development systems (e.g. hardware or mechanics)[3, 15, 17, 19, 27, 33, 54, 61]
Possible Solution:	Synchronize development cycles.
Potential Transfer:	- Embed the agile development and establish an interface between the agile and the surrounding processes. Synchronize important dates like the hardware release with the software development release dates.
Identified Challenge:	Long-term predictability is hard to achieve.[5, 7, 23, 28, 46, 52, 58]
Possible Solution:	More responsibilities for lower hierarchy level to get a flexible development.
Potential Transfer:	- Flatten the hierarchy to shift decisions about implementation to lower hierarchy levels. Less communication effort shortens the decision time and therefore the overall time for development.
Identified Challenge:	It is unclear how to manage software reuse and agile development. [4, 7, 10–12, 15, 21–25, 27, 31–33, 35, 42, 44, 45, 47, 49–51, 55, 57–59, 66, 68, 69]
Possible Solution:	New lightweight variability analysis technique for faster analysis.
Potential Transfer:	- Resolve the hierarchies and give responsibilities to lower hierarchy levels. - Set up a lightweight process to do the agile scoping. - Resolve problems with world-wide coordination.
Identified Challenge:	The usage of open source software [6, 9]
Possible Solution:	The usage of open source software could foster competition and innovation.
Potential Transfer:	- Not desired in the automotive domain. Every OEM wants to protect their intellectual property. Open source software is not considered here.
Identified Challenge:	Communication overheads and problems caused by outsourcing and attitudinal (cultural) incompatibilities in agile projects. [3, 6, 8, 12, 23, 28, 35, 43, 61]
Possible Solution:	New lightweight process and a mindset change of employees.
Potential Transfer:	- Increase the amount of in-house development for less communication. - It is not possible to force a mindset change. Sensitize employees and give them enough time to familiarize with the situation. - Define interfaces, first, between traditional SPLs and agile development, as well as, second, between supplier and OEM.

http://dx.doi.org/10.1007/978-3-319-14358-3_4

[20] F. Farahani and R. Ramsin. 2014. *Methodologies for Agile Product Line Engineering: A Survey and Evaluation.* In *Proc. 13th SOMET.* DOI: http://dx.doi.org/10.3233/978-1-61499-434-3-545

[21] M. Galster and P. Avgeriou. 2014. *Supporting Variability Through Agility to Achieve Adaptable Architectures.* In *Agile Software Architecture.* pp. 139–159. DOI: http://dx.doi.org/10.1016/B978-0-12-407772-0.00005-8

[22] Y. Ghanam, D. Andreychuk, and F. Maurer. 2010. *Reactive Variability Management in Agile Software Development.* In *Proc. AGILE Conf.*. pp. 27–34. DOI: http://dx.doi.org/10.1109/AGILE.2010.6

[23] Y. Ghanam and F. Maurer. 2008. *An Iterative Model for Agile Product Line Engineering.* In *The SPLC Doctoral Symposium.*

[24] Y. Ghanam and F. Maurer. 2010. *Extreme Product Line Engineering - Refactoring for Variability: A Test-Driven Approach.* In *Agile Processes in SE and Extreme Programming*. In Lecture Notes in Business Information Processing, Vol. 48. pp. 43–57. DOI:http://dx.doi.org/10.1007/978-3-642-13054-0_4

[25] Y. Ghanam, F. Maurer, P. Abrahamsson, and K. Cooper. 2009. *A report on the XP workshop on agile product line engineering.* In *ACM SIGSOFT SE Notes* Vol. 34, p. 25. DOI:http://dx.doi.org/10.1145/1598732.1598754

[26] Y. Ghanam, S. Park, and F. Maurer. 2008. *A Test-Driven Approach to Establishing & Managing Agile Product Lines.* In *Proc. 12th SPLC.* pp. 151–156.

[27] L. Gren, R. Torkar, and R. Feldt. 2014. *Work Motivational Challenges Regarding the Interface between Agile Teams and a Non-Agile Surrounding Organization: A Case Study.* In *Proc. 26th ECRTS.* pp. 11–15. DOI:http://dx.doi.org/10.1109/AGILE.2014.13

[28] G. K. Hanssen. 2011. *Agile software product line engineering: Enabling factors.* In *Software: Practice and Experience* Vol. 41, 883–897. DOI:http://dx.doi.org/10.1002/spe.1064

[29] K. Hayashi, M. Aoyama, and K. Kobata. 2015. *A Concurrent Feedback Development Method and Its Application to Automotive Software Development.* In *Proc. 2015 APSEC.* pp. 362–369. DOI:http://dx.doi.org/10.1109/APSEC.2015.54

[30] H. Holdschick. 2012. *Challenges in the evolution of model-based software product lines in the automotive domain.* In *4th Int. Workshop.* pp. 70–73. DOI:http://dx.doi.org/10.1145/2377816.2377826

[31] O. Hummel and C. Atkinson. 2007. *Supporting Agile Reuse Through Extreme Harvesting.* In *Agile processes in SE and extreme programming.* Lecture Notes in Computer Science, Vol. 4536. pp. 28–37. DOI:http://dx.doi.org/10.1007/978-3-540-73101-6_5

[32] H. Jonsson, S. Larsson, and S. Punnekkat. 2012. *Agile Practices in Regulated Railway Software Development.* In *Proc. ISSREW.* pp. 355–360. DOI:http://dx.doi.org/10.1109/ISSREW.2012.80

[33] M. Kaisti, V. Rantala, T. Mujunen, S. Hyrynsalmi, K. Könnölä, T. Mäkilä, and T. Lehtonen. 2013. *Agile methods for embedded systems development - a literature review and a mapping study.* In *EURASIP Journal on Embedded Systems* Vol. 2013, DOI:http://dx.doi.org/10.1186/1687-3963-2013-15

[34] B. Katumba and E. Knauss. 2014. *Agile Development in Automotive Software Development: Challenges and Opportunities.* In *Proc. 15th PROFES.* pp. 33–47. DOI:http://dx.doi.org/10.1007/978-3-319-13835-0_3

[35] M. Kircher and P. Hofman. 2012. *Combining systematic reuse with Agile development.* In *Proc. 16th SPLC.* p. 215. DOI:http://dx.doi.org/10.1145/2362536.2362566

[36] B. Kitchenham. 2007. *Guidelines for performing Systematic Literature Reviews in SE.*

[37] R. Kurmann. 2006. *Agile SPL-SCM Agile Software Product Line Configuration and Release Management.* http://www.lsi.upc.edu/events/aple/Kurmann.pdf

[38] A. Leitner and C. Kreiner. 2011. *Software Product Lines - An Agile Success Factor?* In *Systems, Software and Service Process Improvement.* Communications in Computer and Information Science, Vol. 172. pp. 203–214. DOI:http://dx.doi.org/10.1007/978-3-642-22206-1_18

[39] A. Leitner, R. Mader, Ch. Kreiner, Ch. Steger, and R. Weiß. 2011. *A development methodology for variant-rich automotive software architectures.* In *e & i Elektrotechnik und Informationstechnik* Vol. 128, pp. 222–227. DOI:http://dx.doi.org/10.1007/s00502-011-0001-0

[40] M. Lindvall, D. Muthig, A. Dagnino, C. Wallin, M. Stupperich, D. Kiefer, J. May, and T. Kahkonen. 2004. *Agile software development in large organizations.* In *Computer* Vol. 37, pp. 26–34. DOI:http://dx.doi.org/10.1109/MC.2004.231

[41] P. Manhart and K. Schneider. 2004. *Breaking the ice for agile development of embedded software: an industry experience report.* In *Proc. 26th ICSE.* pp. 378–386. DOI:http://dx.doi.org/10.1109/ICSE.2004.1317460

[42] J. Martinez, J. Díaz, J. Pérez, and J. Garbajosa. 2009. *Software Product Line Engineering Approach for Enhancing Agile Methodologies.* In *Agile Processes in SE and Extreme Programming.* Lecture Notes in Business Information Processing, Vol. 31. pp. 247–248. DOI:http://dx.doi.org/10.1007/978-3-642-01853-4_52

[43] A. Martini, L. Pareto, and J. Bosch. 2012. *Enablers and inhibitors for speed with reuse.* In *Proc. 16th SPLC.* p. 116. DOI:http://dx.doi.org/10.1145/2362536.2362554

[44] J. D. McGregor. 2008. *Agile Software Product Lines - A Working Session.* In *Proc. 12th SPLC.* p. 364. DOI:http://dx.doi.org/10.1109/SPLC.2008.60

[45] J. D. McGregor. 2008. *Mix and Match.* In *The Journal of Object Technology* Vol. 7, p. 7. DOI:http://dx.doi.org/10.5381/jot.2008.7.6.c1

[46] A. Metzger and K. Pohl. 2014. *Software product line engineering and variability management: achievements and challenges.* In *Proc. FOSE 2014.* pp. 70–84. DOI:http://dx.doi.org/10.1145/2593882.2593888

[47] K. Mohan, B. Ramesh, and V. Sugumaran. 2010. *Integrating Software Product Line Engineering and Agile Development.* In *IEEE Software* Vol. 27, pp. 48–55. DOI:http://dx.doi.org/10.1109/MS.2010.31

[48] G. S. Neves and P. Vilain. 2013. *Reactive Variability Realization with Test Driven Development and Refactoring Engineering.* In *Proc. 26th SEKE.* pp. 100–105.

[49] M. A. Noor, R. Rabiser, and P. Grünbacher. 2008. *Agile product line planning: A collaborative approach and a case study.* In *Journal of Systems and Software* Vol. 81, pp. 868–882. DOI:http://dx.doi.org/10.1016/j.jss.2007.10.028

[50] P. O'Leary, M. Ali , S. Thiel, and I. Richardson. 2007. *Product derivation process and agile approaches:exploring the integration potential.* In *2nd IFIP CEE Conf. on SE Techniques.*

[51] P. O'Leary, F. McCaffery, I. Richardson, and S. Thiel. 2009. *Towards Agile Product Derivation in Software Product Line Engineering.*

[52] P. O'Leary, F. McCaffery, S. Thiel, and I. Richardson. 2012. *An agile process model for product derivation in software product line engineering.* In *Journal of Software: Evolution and Process* Vol. 24, pp. 61–571. DOI:http://dx.doi.org/10.1002/smr.498

[53] H. Holmström Olsson, J. Bosch, and H. Alahyari. 2013. *Customer-Specific Teams for Agile Evolution of Large-Scale Embedded Systems.* In *Proc. 39th SEAA.* pp. 82–89.

[54] H. Holmström Olsson, J. Bosch, and H. Alahyari. 2013. *Towards R&D as Innovation Experiment Systems: A Framework for Moving Beyond Agile Software Development.* In *Proc. of the IASTED Multi-Conf..* DOI:http://dx.doi.org/10.2316/P.2013.796-008

[55] R. F. Paige, X. Wang, Z. R. Stephenson, and P. J. Brooke. 2006. *Towards an Agile Process for Building Software Product Lines.* In *Extreme Programming and Agile Processes in SE.* pp. 198–199. DOI:http://dx.doi.org/10.1007/11774129_24

[56] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. 2008. *Systematic Mapping Studies in SE.* In *Proc. 12th EASE.* pp. 68–77.

[57] P. Pohjalainen. 2011. *Bottom-up Modeling for a Software Product Line: An Experience Report on Agile Modeling of Governmental Mobile Networks.* In *Proc. 15th SPLC.* pp. 323–332. DOI:http://dx.doi.org/10.1109/SPLC.2011.48

[58] M. Raatikainen, K. Rautiainen, V. Myllärniemi, and T. Männistö. 2008. *Integrating product family modeling with development management in agile methods.* In *Proc. 1st Int. Workshop on Software Development Governance.* p. 17. DOI:http://dx.doi.org/10.1145/1370720.1370728

[59] A. Jr. Santos and V. Jr. Lucena. 2010. *SCRUMPL - Software Product Line Engineering with Scrum.* In *Proc. 5th ENASE.* pp. 239–244. DOI:http://dx.doi.org/10.5220/0003038302390244

[60] A. Schloßer, J. Schnitzler, T. Sentis, and J. Richenhagen. 2016. *Agile processes in automotive industry - Efficiency and quality in software development.* In *16. Int. Stuttgarter Symposium.* pp. 489–503. DOI:http://dx.doi.org/10.1007/978-3-658-13255-2_35

[61] M. Shen, W. Yang, G. Rong, and D. Shao. 2012. *Applying agile methods to embedded software development: A systematic review.* In *Proc. 2nd SEES.* pp. 30–36. DOI:http://dx.doi.org/10.1109/SEES.2012.6225488

[62] J. Taiber and J. D. McGregor. 2014. *Efficient engineering of safety-critical, software-intensive systems.* In *Proc.ICCVE.* pp. 836–841. DOI:http://dx.doi.org/10.1109/ICCVE.2014.7297669

[63] R. Y. Takahira, L. R. Laraia, F. A. Dias, S. Y. Abraham, P. T. S. Nascimento, and A. S. Camargo. 2014. *Scrum and Embedded Software development for the Automotive Industry.* In *Proc. PICMET'14.* pp. 2664–2672.

[64] S. Thiel, M. A. Babar, G. Botterweck, and L. O'Brien. 2009. *Software Product Lines in Automotive Systems Engineering.* *SAE Int. Journal of Passenger Cars - Electronic and Electrical Systems* Vol. 1, pp. 531–543. DOI:http://dx.doi.org/10.4271/2008-01-1449

[65] K. Tian. 2014. *Adding More Agility to Software Product Line Methods.* In *International Journal of Knowledge and Systems Science* Vol. 5, pp. 17–34. DOI:http://dx.doi.org/10.4018/ijkss.2014100102

[66] K. Tian and K. Cooper. 2006. *Agile and Software Product Line Methods: Are They So Different?*

[67] P. Trinidad, D. Benavides, A. Durán, A. Ruiz-Corts, and M. Toro. 2008. *Automated error analysis for the agilization of feature modeling.* In *Journal of Systems and Software* Vol. 81, pp. 883–896. DOI:http://dx.doi.org/10.1016/j.jss.2007.10.030

[68] P. Trinidad, D. Benavides, A. Ruiz-Corts, . Segura, and M. Toro. 2006. *Explanations for Agile Feature Models.* http://www.lsi.upc.edu/events/aple/

[69] S. Urli, M. Blay-Fornarino, P. Collet, and S. Mosser. 2012. *Using composite feature models to support agile software product line evolution.* In *Proc. 6th Int. Workshop on Models and Evolution*, pp. 21–26. DOI:http://dx.doi.org/10.1145/2523599.2523604

[70] R. Valade. *The Big Projects Always Fail: Taking an Enterprise Agile.* In *Agile 2008 Conf..* pp. 148–153. DOI:http://dx.doi.org/10.1109/Agile.2008.63

[71] S. Weber. April 2015. *Agile in Automotive - State of Practice 2015.* www.kuglermaag.com/agile2015

[72] C. Wohlin. 2014. *Guidelines for snowballing in systematic literature studies and a replication in SE.* In *Proc. EASE 2014.* DOI:http://dx.doi.org/10.1145/2601248.2601268

[73] L. Wozniak and P. Clements. 2015. *How automotive engineering is taking product line engineering to the extreme.* In *Proc. 19th SPLC.* pp. 327–336. DOI:http://dx.doi.org/10.1145/2791060.2791071