An Assessment Model to Foster the Adoption of Agile Software Product Lines in the Automotive Domain

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*—A software product line is commonly used for the software development in large automotive organizations. A strategic reuse of software is needed to handle the increasing complexity of the development and to maintain the quality of numerous software variants. However, the development process needs to be continuously adapted at a fast pace to satisfy the changing market demands. Introducing agile software development methods promise the flexibility to react on customers' change requests and market demands to deliver high quality software. Despite this need, it is still challenging to combine agile software development and product lines. The maturity of an agile adoption is often hard to determine. Assessing the current situation regarding the combination is a first step towards a successful inclusion of agile methods into automotive software product lines. Based on an interview study with 16 participants and a literature review, we build the so-called ASPLA Model allowing self-assessments within the team to determine the current state of agile software development in combination with software product lines. The model comprises seven areas of improvement and recommends a possibility to improve the current status.**

***Keywords*— agile software development, software product lines, process maturity framework, software process improvement, automotive domain, embedded software development, ASPLA Model.**

1. Introduction

Software is present in all areas of our daily life. In 2016, approximately 14% of Germans used wearables [1], such as smart watches or fitness trackers to monitor their movements and observe health status. According to Thomas [2], about 86% of the world's population own a smartphone. Nowadays, every smartphone can remotely control things, such as smart homes and cars from all around the world. The cars we drive are getting smarter and are connected to the environment. All over the world, people are craving for new technological enhancements and electronic devices. This desire can be observed in all industry sectors and leads to complex software to satisfy requirements from customers and other marked demands.

The increasing complexity of software can be addressed by a strategic reuse, to manage the development and to maintain the quality of numerous customized software variants. Software product lines are a software paradigm for systematic software reuse and commonly used in the automotive software development [3]. In the automotive embedded development it is necessary to manage the high number of different software variants that meet different requirements across multiple markets [4], while simultaneously maintaining the quality of the software.

According to Wozniak et al. [5], the automotive domain is the most challenging environment for systems and software product line engineering. Millions of different software variants exist, whereby each one of them comprises of a large complexity. The complexity is based on the large number of variation points within the product [6]. According to the high numbers of different software variants, it is common to reuse software parts. Software product lines help to manage changes, coordinate the worldwide software development, and increase the software quality by reuse. Furthermore, the software variants enable a customization and individualization of customers' requirements and generate more value for the core business.

The development processes are seen as too slow, to keep pace with the fast changing market demands [7], [8]. Therefore, the development processes need to be redesigned in a way that it is possible to learn and adapt continuously at a fast pace. The use of selected agile practices are promising approaches to address these new challenges. Since 2001, agile software development methods promise an improved software development with a faster time to market and an increasing speed of learning. Furthermore, agile development practices offer the possibility to react on changing requirements and to refine the final software during the development process. The adaption of agile practices within the automotive domain is therefore a possible way to keep pace with fast changing market demands [9].

Current software development in the automotive domain is heavily structured by standardized processes. Process assessments are used to evaluate the processes of the organizational unit against a predefined process assessment model. The most popular standards in the automotive domain are CMMI [10] and Automotive SPICE (ASPICE) [11].

Assessing the current status of the development is a prerequisite for a successful combination of agile methods and software product lines in the automotive domain. We identified the need for an adjusted assessment model, addressing Agile Software Product Lines in the Automotive Domain (ASPLA Model) [13]. The ASPLA Model helps to assess the current status and proposes recommendations to improve the agile software development. This paper aims to answer the following research question:
Which aspects need to be considered for an adjusted assessment model that assesses an organization's current situation regarding agile software development and software product lines?

The paper is separated in two parts. First, it presents the basics of the ASPLA Model that allows automotive organizations to assess the current status for the combination of agile software development and software product lines. The identified areas are presented and described in detail. Second, the outcomes that are addressed in the ASPLA Model are validated by a correlation of the ASPLA Model and the *reference model for product line engineering and management* (ISO 26550) [14].

The remainder of this paper is organized as follows. In section 2, we present related work. Section 3, presents the research approach. The ASPLA Model is presented in section 4, followed by the conclusion in section 5. The paper gives an outlook on future work in section 6.

2. Relation to existing theories and work

This section is divided into two parts. The first part presents existing assessment models that focus on software reuse strategies and are currently in use in the automotive domain. The second part investigates publications that introduce successful implementations of combinations of agile software development and software product lines. These insights can be tailored and transfered to the automotive domain.

2.1. Automotive assessment models

Different context specific standards have evolved for assessments over the past. The two well-known standards in the automotive domain are ASPICE [11] and CMMI [10]. Both standards define methods to evaluate complete process models and organizations [12]. For automotive embedded software and systems, the Automotive SPICE (ASPICE) process assessment model is applied. ASPICE is maintained by the German Association of the Automotive Industry [11]. CMMI [10] is a collection of best practices that supports organizations to improve their processes. CMMI focuses on activities for developing products to meet the needs of customers [10]. Neither CMMI [10] nor ASPICE [11] focus explicitly on the combination of agile development practices and software product lines. In contrast, the ASPLA model presented in this publication addresses the issue. We present the tailored assessment model, focusing explicitly on agile development and software product lines in the automotive domain.

Jasmine and Vasantha [15] define the *Reuse Capability Maturity Model* (RCMM). This model focuses on a well planned and controlled reuse oriented software development. The RCMM is based on CMMI [10]. It comprises maturity levels that denote an achieved level in the evolution to a mature reuse process [15]. However, the model does not focus on agile software development and has not yet been evaluated [15]. Therefore, it is unknown if this model is also applicable in the automotive domain. In contrast to the presented model of Jasmine and Vasantha [15], our assessment approach is tailored especially to the automotive domain. It retains the established benefits resulting from the structured software reuse and enrich the development by suitable agile practices and methods.

Another assessment model for the automotive domain is the *Agile, Hybrid Assessment Method for the Automotive Industry (AHAA)* [16] This assessment model considers safety-critical development and aims at achieving a high software quality to ensure high reliability and maturity of software. The AHAA is a low-overhead assessment method specifically for small-to-medium sized organizations. The assessment suggests agile based improvement solutions combined with CMMI and ASPICE. However, the AHAA does not focus on software reuse in particular. Therefore the applicability of the AHAA to assess agile software product lines in the automotive domain is limited.

Hantke [17] presents a development process that includes elements from SCRUM [18] and parts of the standard SPICE [19]. With his approach, requirements get more transparent and the team communications is improved. Furthermore, he maps SPICE practices to SCRUM activities. He presents a pilot test for the combined approach. In contrast to our approach, Hantke [17] did not focus on software product line and a strategic software reuse.

2.2. Agile software product lines in automotive

The published literature does not provide significant information or approaches on how to adapt existing agile software development approaches to software product line development in the automotive domain [20]. However, the insights from the publications can be used, tailored and transferred to the automotive domain.

Thiel et al. [4] analyze the combination of agile software development and plan-driven processes. They suggest to introduce context specific agile practices and methods to automotive systems engineering [4]. Furthermore, they mention that rigid quality and safety requirements could benefit from the combination [4]. The Kugler and Maag study ”Agile in Automotive Survey” confirms this approach [21]. As identified by the survey, many companies only introduce single agile practices and methods into the development processes. Schloßer et al. [22] identify the necessity of shorter development cycles for multiple software variants, introduced by a high cost pressure. They mention that incremental software deliveries in a shorter time and a quick response from customer could be achieved [9], [22].

However, none of the presented approaches focus on a strategic software reuse strategy or a software product line in detail [4], [9], [21], [22]. Different models are presented within the literature in order to integrate agile elements into the automotive domain in a more structured way.

To enhance cooperation between different organizations, the concurrent *Feedback Loop Model* introduces feedback loops [23]. This model sets up a new architect role to manage the communication between the organizations. The improved communication leads to a shorter development time and ensures internal quality [23]. The feedback loops improve the collaboration between the organizations effectively. Taiber and McGregor [24] introduce the *System Architecture Virtual Integration* (SAVI). SAVI supports agile development by the introduction of Continuous Integration in the software development process. Integration is realized by the use of a virtual integration environment [24].

Another model that supports the handling of complexity, is the *Mega Scale Software Product Line Engineering* (MS-SPLE) approach. The MS-SPLE approach can be applied in large projects with complex products and feature variations. However, agile software development practices are not taken into account within MS-SPLE [5].

None of the presented models [5], [23], [24] focus on a combination of agile software development and software product lines. In total, two different trends could be identified in the literature:

- An overall accepted combination of agile software development and software product lines for the automotive domain does not exist. The published literature does not show any recommendations to use a comprehensive set of agile elements and practices in combination with a software product line in the automotive domain. However, the presented methods and processes include interesting new concepts that could be tailored for a new automotive specific model.
- Existing assessment models do not explicitly consider agile software development and software product lines in the automotive domain. However, few hints and notes to consider agile elements are given (cf. CMMI [10]).

3. Research approach

Within this publication, we provide a description of the ASPLA Model. It is based on findings from our previous work: First, the ASPLA Model is based on the results of a literature review [20]. This study identified possible solutions which could be adopted for the automotive domain. Second, the ASPLA Model comprises the outcomes of an interview study [13] with practitioners from the automotive domain. The interviewed practitioners gave recommendations on the combination of agile development and software product lines in the automotive domain and identified challenges that hinder a combination. We address these insights in the definition of the ASPLA Model. Furthermore, current standards such as CMMI [10], ASPICE [11] and ISO 26550 [14] were evaluated and compared to the ASPLA Model. To validate the ASPLA Model, the following two methods were used:

- *Similarity:* The used structure within the ASPLA Model is ajar to the defined structure of the ASPICE [11] standard. Due to the fact that ASPICE is commonly used in the automotive domain, the introduction of the ASPLA Model is presumed to be more acceptable.
- *Validity by Construction:* The ASPLA Model is compared and matched to the ASPICE [11] standard and the *reference model for product line engineering and management* (ISO 26550) [14]. This ensures that benefits introduced by the use of the software product line, as stated in the standard, are not suspended.

4. Findings

This section is divided into the two parts that focus on the two construction methods of the ASPLA Model mentioned in section 3. Sections 4.1 to 4.4 describe the ASPLA Model and set it into correlation with the structure from ASPICE [11]. Section 4.5 examines the Standard ISO 26550 [14] and matches it to the ASPLA Model.

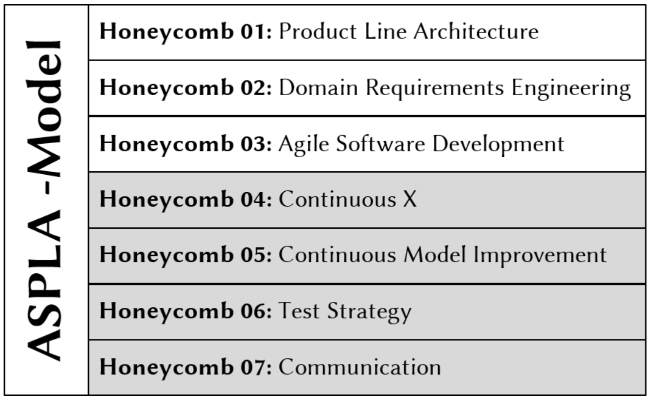
4.1. ASPLA Model: Introduction

We identified seven important areas which need to be considered in an assessment for the combination of agile development and software product lines. The areas are (1) Product Line Architecture, (2) Domain Requirements Engineering, (3) Agile Software Development, (4) ContinuousX, (5) Continuous Model Improvement, (6) Test Strategy, and (7) Communication. The first three areas that directly deal with the agile software product line (*Product Line Architecture, Domain Requirements Engineering and Agile Software Development*) are obvious to consider. However, there exist further areas that are often forgotten while introducing agile into an organization. Therefore, this publication does not focus on the obvious issues that need to be considered, such as agile development and software product lines. This publication focus on the areas *ContinuousX, Continuous Model Improvement, Test Strategy,* and *Communication*. However, it is important to consider all areas in total, to build the basis for the ASPLA Model. The areas can be used by developers, as a guideline for self assessment.

4.2. ASPLA Model: The Honeycombs

In the terminology of the ASPLA model, each identified area is called a “Honeycomb” (cf. Figure 1). The areas are denoted as “Honeycombs” to make a very clear distinction of the ASPLA Model from ASPICE [11]. While ASPICE focuses on processes and outcomes, ASPLA extends the process view and comprises a product-oriented view. For the combination of agile software development and software product lines, it is important that the product is suitable and can cope with the development. Therefore, special product properties are taken into account. Without these product properties, a combination of agile software development and software product line is not achievable.

Each Honeycomb is described by the Honeycomb name and the Honeycomb purpose.



ASPLA -Model

Honeycomb 01: Product Line Architecture

Honeycomb 02: Domain Requirements Engineering

Honeycomb 03: Agile Software Development

Honeycomb 04: Continuous X

Honeycomb 05: Continuous Model Improvement

Honeycomb 06: Test Strategy

Honeycomb 07: Communication

Figure 1. The ASPLA Model: Identified areas of activity (Honeycombs)

Honeycomb 01. The purpose of *Product Line Architecture* is to provide a suitable software architecture to enable the implementation of several software variants for different products with a high degree of software reuse.
Honeycomb 02. The purpose of *Domain Requirements Engineering* is to identify the reuse assets that should be developed in a software product line. This includes the identification of products and features that should be part of the product line and the definition of common and variable features.
Honeycomb 03. The purpose of *Agile Software Development* is to react faster on customer needs and legal constraints to reduce the time to market for innovative feature upon a simultaneous increase of software quality.
Honeycomb 04. The purpose of *ContinuousX* is to continuously execute tasks which lead to a more stable, compliant and better product.
Honeycomb 05. The purpose of *Continuous Model Improvement* is to continuously reflect on the ASPLA Model and improve the interaction of the assessment result and the suggested improvement for the software development process.
Honeycomb 06. The purpose of *Test-Strategy* is to provide an environment to verify the correct behavior and ensure the software quality for various software variants which are developed in a fast development pace within the software development.
Honeycomb 07. The purpose of *Communication* is to verify the communication of all participating roles to avoid knowledge silos and to react on customer needs faster.

4.3. ASPLA Model: The Outcomes

Each Honeycomb defines outcomes that need to be achieved at a sufficient level. In contrast to ASPICE [11], the the outcomes for the ASPLA Model are divided into *Process related Outcomes* and *Product related Outcomes*.

In total, the ASPLA Model consists of 52 Process related Outcomes and 12 Product related Outcomes. Honeycomb 05 “Continuous Model Improvement” and Honeycomb 07 “Communication” do not require any Product related Outcome.

Product related Outcomes are expected positive results of the development performance which are induced by properties of the product. The ASPLA Model defines requirements for the product under development. An exemplary selection of Product related Outcomes is presented in Table 1 for Honeycomb 04 and Honeycomb 06.

TABLE 1. EXEMPLARY PRESENTATION OF PRODUCT RELATED OUTCOMES FOR HONEYCOMB 04 AND HONEYCOMB 06

Product related Outcomes

ID	Outcome
04 Continuous X	
ProdO_04.01	Software units and software items are integrated up to a complete integrated software according to the integration strategy.
ProdO_04.02	Software items provide the opportunity to perform verification activities with a high level of automation in regular time intervals.
06 Test Strategy	
ProdO_06.01	For each software unit an associated test case assigned.

Process related Outcomes are expected positive results of the process performance which are related to the proper implementation of the development process. Exemplary, some Process related Outcomes are presented in Table 2 for Honeycomb 04, Honeycomb 05, Honeycomb 06 and Honeycomb 07. Table 2 is not comprehensive.

4.4. ASPLA Model: Honeycomb attributes

Furthermore, the ASPLA Model provides Honeycomb attributes, which include *Work Products*, *Base Practices*, and *Product Attributes*. The internal structure of a Honeycomb is shown in Figure 2.

The Honeycomb attributes can be used by the assessor to define the Honeycomb maturity. During an assessment, the Honeycomb attributes are supposed to be a quickly accessible information source. They represent an exemplary implementation, the assessor can use. The relations between the purpose, the outcomes and the Honeycomb attributes are defined as follows:

1) *The Honeycomb-Outcomes* are the results of a successful implementation to fulfill the *Honeycomb-Purpose*.

TABLE 2. EXEMPLARY PRESENTATION OF PROCESS RELATED OUTCOMES FOR HONEYCOMB 04 - HONEYCOMB 07

Process related Outcomes

	ID	Outcome
04 Continuous X	ProcO_04.03	Continuous software tests are applied and failures in the software are identified earlier.
	ProcO_04.04	Continuous delivery to internal customers is applied and supported by the development process.
	ProcO_04.06	The software development process ensures that legal constraints are abided.
05 Model Improvement	ProcO_05.01	Suggestions to improve the software development process are proposed by the ASPLA-Model.
	ProcO_05.02	The efficiency of the development process is continuously evaluated.
	ProcO_05.03	Best practices from the software development process are reduced into the ASPLA-Model.
	ProcO_05.04	The ASPLA-Model is reviewed in a regular timeframe.
06 Test Strategy	ProcO_06.03	A scalable test strategy for different test hierarchies, software variants, systems and real cars is realized.
	ProcO_06.04	The individual priority of test cases is determined in a regular time period by the test strategy and suitable test cases are automatically executed.
	ProcO_06.05	Consistency and bidirectional traceability is established.
07 Communication	ProcO_07.01	All involved parties are participate in the development process directly.
	ProcO_07.03	Fast feedback channels for faster learning loops are established.
	ProcO_07.09	Direct communication with the customer is enhanced.

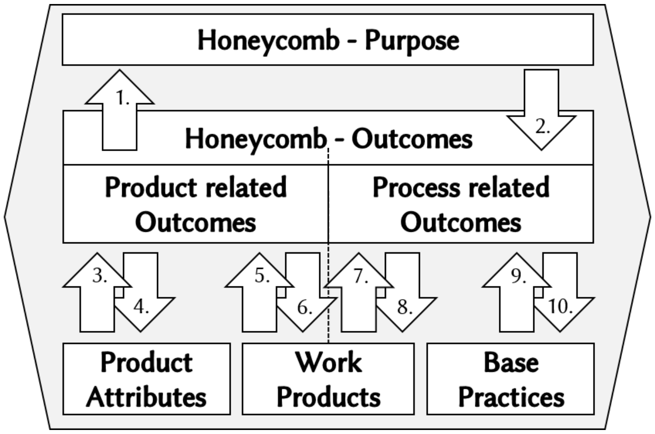


Figure 2. Honeycomb - Internal structure

2) The *Honeycomb-Purpose* is characterized by the *Honeycomb-Outcomes*.
3) *Product Attributes* indicate a successful achievement of *Product related Outcomes*.
4) *Product related Outcomes* are supported by the *Product Attributes*.
5) *Work Products* indicate a successful achievement of *Product related Outcomes*.
6) *Product related Outcomes* affect the *Work Products*.
7) *Work Products* indicate a successful achievement of *Process related Outcomes*.
8) *Process related Outcomes* lead to *Work Products*.
9) *Base Practices* indicate a successful achievement of *Process related Outcomes*.
10) *Process related Outcomes* are supported by *Base Practices*.

Product Attributes define the characteristics of the product to be developed and represent product oriented Honeycomb attributes. For a combination of an agile software product line, the implementation of a modular architecture, standardized interfaces between software units or the use of backlogs to store features are important Product Attributes. If Product Attributes are not fulfilled, it is challenging to enable a combination of agile software development and software product lines. Table 3 summarizes the associated Product Attributes for the Product related Outcomes (cf. Table 1).

TABLE 3. ASSOCIATED PRODUCT ATTRIBUTES FOR HONEYCOMB 04 AND HONEYCOMB 06 (CF. TABLE 1)

Product Attributes	Outcome
PA_ProdO_04.01: An integration strategy shall be implemented. Ensure that software units and software items are integrated up to a complete integrated software according to the integration strategy, by means of integration server and virtual integration.	ProdO_04.01
PA_ProdO_04.02: Continuous verification tasks shall be automated. Ensure that the software units are tested for compliance, by means of software items which are capable to undergo highly automated verification tasks.	ProdO_04.02
PA_ProdO_06.01: Test case shall be implemented for each software units. Ensure that all software units, especially critical software units do have an associate test case, which consists of all test steps to be compliant with the law, by means of a traceable 1 to 1 mapping between test case and software unit.	ProdO_06.01

Base Practices represent activity-oriented indicators. They indicate a successful achievement of the Process related Outcomes. The definition of the ASPLA Model maps one Base Practice on to one outcome.

The associated Base Practices for the Process related Outcomes (cf.Table 2) are presented in Table 4.

Work products represent result oriented indicators that indicate a successful achievement of Product related Outcomes and Process related Outcomes. The ASPLA Work Products are ajar to the Work Products defined in ASPICE [11]. ASPICE Work Products are currently valid for assessments in the automotive domain and shall remain

TABLE 4. Associated Base Practices for Honeycomb 04, Honeycomb 05, Honeycomb 06 and Honeycomb 07 (cf. Table 2)

	Base Practices	Outcome
04 Continuous X	**BP_ProcO_04.03:** Apply continuous tests for software. Ensure that software failures are identified by timed and automated software tests.	ProcO_04.03
	BP_ProcO_04.04: Set up continuous delivery. Ensure that software is delivered to (internal) customers in a fast pace and the infrastructure to deliver the software is available.	ProcO_04.04
	BP_ProcO_04.06: Set up continuous compliance. Ensure that each software release is (potentially) compliant.	ProcO_04.06
05 Model Improvement	**BP_ProcO_05.01:** Propose suggestion to improve the development process. Ensure that suggestions are available according to the evaluation and assessment result given by the ASPLA-Model.	ProcO_05.01
	BP_ProcO_05.02: Monitor development efficiency. Ensure that metrics are selected to measure the efficiency and the implications on the process when suggestions for improvement are introduced.	ProcO_05.02
	BP_ProcO_05.03: Reduce Best Practices. Ensure that best practices from efficient software development processes are reduced into the suggestions given by the ASPLA-Model if they are not included so far.	ProcO_05.03
	BP_ProcO_05.04: Review the ASPLA-Model. Ensure that the ASPLA-model is reviewed in a defined timeframe.	ProcO_05.04
06 Test Strategy	**BP_ProcO_06.03:** Implement a scalable test strategy. Ensure that the test strategy covers tests for individual software variants as well as test in the entire systems and in real cars.	ProcO_05.02
	BP_ProcO_06.04: Assign test case priority and run suitable tests cases. Ensure that high prior test cases, such as safety related test cases, are automatically selected and executed.	ProcO_05.03
	BP_ProcO_06.05: Maintain traceability. Ensure that a consistent and bidirectional traceability is established between the software units, the test cases and between test cases and test results.	ProcO_05.04
07 Communication	**BP_ProcO_07.01:** Involve all parties affected by the development. Ensure that all involved parties are involved directly in the development process.	ProcO_07.01
	BP_ProcO_07.03: Establish fast feedback loops. Ensure that fast feedback loops are established to provide a faster learning loop. This includes automatic generated feedback from test systems, feedback from customer, and feedback from all affected parties.	ProcO_07.03
	BP_ProcO_07.09: Involve the customer. Ensure that a direct customer communication is established.	ProcO_07.09

valid in the ASPLA Model. To set up the ASPLA Model, all available Work Products of ASPICE [11] were taken into account, to get a comprehensive overview.

Overall, 48 Work Products were adjusted to complement missing parts from ASPICE regarding agile software development and software product lines in the automotive domain. Therefore, the ASPLA Work Products either extend the content or exclude parts of the ASPICE Work Products. Due to the adjustments, 23 Work Products were assimilated within other ASPLA Work Products. Furthermore, 36 ASPICE Work Products, such as the *Customer manual* and the *Handling and storage guide*, are out of scope or not applicable in the ASPLA Model.

ASPLA Work Products indicate a successful achievement of Product and Process related Outcomes (cf. Figure 2). Selected examples of Work Products are presented in this publication.

A_08-50 Test Specification. In ASPICE [11], the "Test Specification" is a Work Product that comprises the test case specification and the identification of test cases for regression testing. However, to make this Work Product (Test Specification) compatible with agile software development of many software variants, further properties are need to be added. The ASPLA Work Product *A_08-50 Test Specification* requires the introduction of a priority ranking of test case. with this, the test strategy can decide which test are necessary to select for automated testing. Safety critical tests shall get a higher priority and shall be preferred in test case selection. Figure 3 shows a simplified representations how Work Products are connected to the outcomes. Some Work Products relate to one or more Honeycomb outcomes.

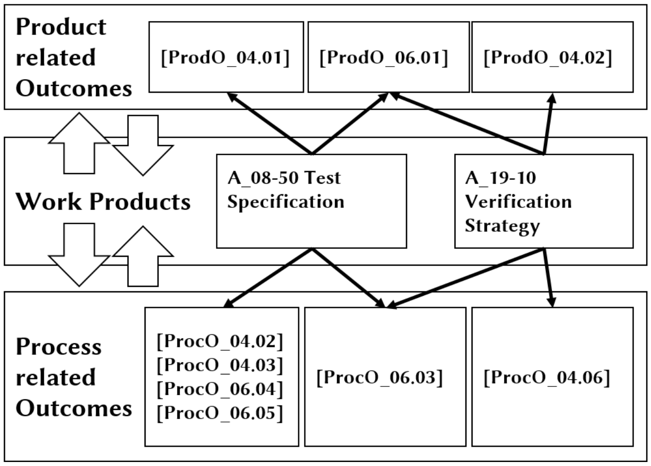


Figure 3. Exemplary presentation of Work Products and the relation to the outcomes

A_19-10 Verification Strategy. The second example within this publication is the Work Product *A_19-10 Verification Strategy*. In ASPICE [11] the "Verification Strategy" describes verification methods, techniques, and tools. The ASPLA Work Product comprises the strategy for a sufficient test coverage for all products and software variants, to fulfill legal requirements.

It is mandatory for the assessor to consider the whole ASPLA Model and not only parts of it. All honeycombs are connected by shared work Products. Figure 4 shows a small section of the dependency structure of the ASPLA Model.

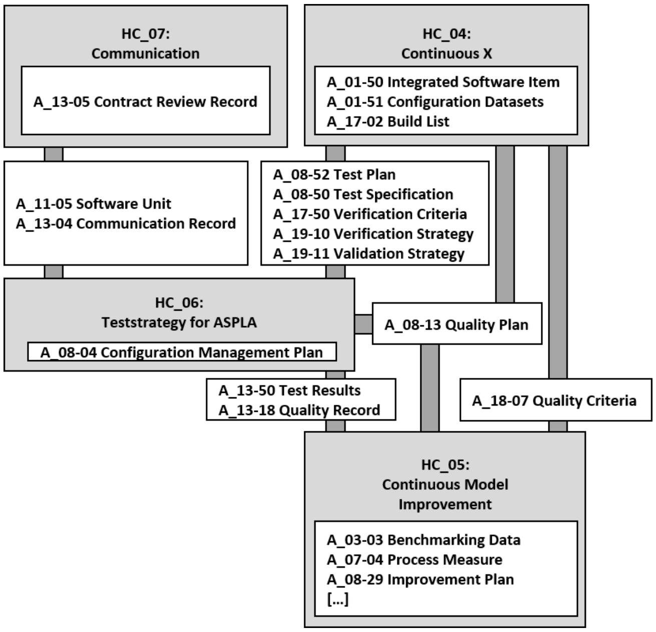

HC_07:
Communication
A_13-05 Contract Review Record
HC_04:
Continuous X
A_01-50 Integrated Software Item
A_01-51 Configuration Datasets
A_17-02 Build List
A_11-05 Software Unit
A_13-04 Communication Record
A_08-52 Test Plan
A_08-50 Test Specification
A_17-50 Verification Criteria
A_19-10 Verification Strategy
A_19-11 Validation Strategy
HC_06:
Teststrategy for ASPLA
A_08-04 Configuration Management Plan
A_08-13 Quality Plan
A_13-50 Test Results
A_13-18 Quality Record
A_18-07 Quality Criteria
HC_05:
Continuous Model
Improvement
A_03-03 Benchmarking Data
A_07-04 Process Measure
A_08-29 Improvement Plan
[...]

Figure 4. Exemplary presentation of Work Products and the relation to the outcomes

4.5. Validation

The Elements in the ASPLA model are mapped to parts of relevant and current automotive domain standards. This ensures that the model complies with those standards and verifies that the ASPLA Model is suitable for the automotive domain.

Therefore, the ASPLA Model was compared to ISO 26550 [14]. Features of the product line development are considered and relevant parts are represented within the ASPLA Model. The ISO 26550 [14], defines the reference model for product line engineering and management. Special care was taken to include all essential components of the ISO 26550 [14] standard in the ASPLA Model. As mentioned in ISO 26550 [14], product line organizations should design their structures and processes to address specific issues. Many specific issues such as Application engineering, Domain asset, Domain engineering, are addressed by the first 3 Honeycombs. A complete list of all issues can be found in section 4.3 within ISO 26550 [14].

Within this publication, we focus on the specific issues of (1) Measurement and tracking, (2) Traceability, and (3) Verification and validation.

(1) Measurement and tracking. Measurement tasks in product line engineering and the management of the product line itself, are complex. Due to the separation of domain engineering and application engineering life cycles, the data collection, the measurement and tracking needs to be synchronized. Furthermore, the organizational and technical management of the product line needs to be considered to receive a proper measurement result.

(2) Traceability. The development in a software product line is typically knowledge-intensive and a lot of collaboration and coordination. It is important to trace and manage the knowledge to control the complexity of the overall software product line and the development of single variants. Traceability helps to keep track of decisions regarding the development.

(3) Verification and validation. Verification and validation confirms that the requirements for all domain assets and member products are fulfilled. Verification and validation in product line context must consider all software variants and are therefore fundamentally different from the single-system engineering context.

Table 5 summarizes the coverage of the ISO 26550 process specific issues [14] and presents a mapping to the presented outcomes in Table 2.

TABLE 5. EXEMPLARY MAPPING FROM ISO 26550 TO THE ASPLA-MODEL

Structures and Processes	addressed in:
Measurement and tracking	ProcO_05.01, ProcO_05.02
Traceability	ProcO_06.05, [...]
Verification and validation	ProcO_04.06, ProcO_06.01, ProcO_06.02, ProcO_06.03, [...]

5. Conclusion

The combination of agile software development and software product lines in the automotive domain is seen as a promising approach. With this approach, a shorter time to market and a faster learning loop about the maturity of the software could be achieved. These perceived benefits are mentioned in existing literature ([22], [25], [26]). However, the literature often recommends to introduce single agile practices into plan-driven processes in one particular context.

A holistic approach to combine agile software development and software product lines in the automotive domain is not published in the literature [20]. The current status on the agile adoption within software product line is hard to define. We identify the need for an automotive-specific assessment model in previous research ([13], [20]).

In this paper, we examine the aspects that need to be considered for an adjusted assessment model that assess an organization's current situation regarding agile software development and software product lines.

Several assessment models, such as [10], [11], [15], [16], [17] are in use. However, the assessment models do not focus on agile practices in detail.

We address these insufficient assessment models with the definition of the ASPLA Model. The ASPLA Model comprises the results from a literature review [20], an interview study [27] and the identified challenges [13]. We introduce important areas that need special attention in a assessment focusing on agile software development and software product lines.

Within this publication, we explain important areas, such as the test strategy and the communication structure. The ASPLA Model focuses on a valid test strategy that can cope with the various software variants. Therefore, it assesses if the test strategy is scalable to the number of variants. Furthermore, a proper test strategy shall organize the test cases for each automated test according to the changes in the software and the critically of the change. Continuous Integration is defined as as the "backbone" for the testing process. Communication is another important field to consider. The ASPLA Model recommends a close customer contact. Breaking down the "knowledge silos" and establish an open communication is recommended by the ASPLA Model.

Nevertheless, the content of the ASPLA Model is ajar to ASPICE [11]. This increases the acceptance of the model. For a successful combination, not only a process view needs to be considered. The ASPLA Model comprises a specific product properties from the product under development. To increase the validity of the ASPLA Model, the model comprises elements from CMMI [10] and ISO 26550 [14].

The ASPLA Model can be used as a guideline in an assessment to identify Honeycombs which need to be considered. Due to the adaption to the context of agile software product lines in automotive, it foster an agile introduction more than other assessment models. Furthermore, it is based on best practices what leads to an acceptance of the model. The model was primarily designed for the automotive domain and may not be generalized for other domains. Furthermore, tt cannot be guaranteed that the presented assessment model is the only possible way to introduce an improvement.

6. Outlook and Future Work

For future work, we plan to further evaluate the ASPLA Model. A case study shall verify whether the model addresses important improvement areas in practice and it is easy to use by developers in an self-assessment. Therefore, we will make a comparison between an "ad hoc-assessment" executed by an experienced assessor and one assessment following the guidelines proposed by the ASPLA Model conducted by the Researcher.

For further validation, we will evaluate how well the ASPLA Model identifies and represent the current state of the team under assessment regarding the combination of agile software product lines in the automotive domain.

References

[1] T. Jülicher and M. Delisle, "Step into the circle-a close look at wearables and quantified self," in *Big Data in Context: Legal, Social and Technological Insights*, T. Hoeren and B. Kolany-Raiser, Eds. Cham: Springer International Publishing, 2018, pp. 81–91. [Online]. Available: https://doi.org/10.1007/978-3-319-62461-7_10

[2] D. Thomas, "Cellphone addiction and academic stress among university students in thailand," *INTERNATIONAL FORUM JOURNAL*, vol. 19, no. 2, pp. 80–96, url = http://ojs.aiias.edu/index.php/ojs/article/view/187, 2017.

[3] A. Leitner, R. Mader, C. Kreiner, C. Steger, and R. Weiß, "A development methodology for variant-rich automotive software architectures," *e & i Elektrotechnik und Informationstechnik*, vol. 128, no. 6, pp. 222–227, 2011.

[4] 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.

[5] L. Wozniak and P. Clements, "How automotive engineering is taking product line engineering to the extreme," in *the 19th International Conference*, pp. 327–336.

[6] M. Galster and P. Avgeriou, "Supporting variability through agility to achieve adaptable architectures," in *Agile Software Architecture*, 2014, pp. 139–159.

[7] 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.

[8] U. Eliasson, R. Heldal, J. Lantz, and C. Berger, "Agile model-driven engineering in mechatronic systems - an industrial case study," in *Model-Driven Engineering Languages and Systems*, 2014, vol. 8767, pp. 433–449.

[9] 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.

[10] S. C. P. Team, *Cmmi for development v1.3: Improving processes for developing better products and services*, 2011.

[11] VDA QMC Working Group 13 / Automotive SIG, "Automotive spice process assessment / reference model," 01.11.2017. [Online]. Available: www.automotivespice.com

[12] R. Heimgärtner, A. Solanki, and H. Windl, "Cultural user experience in the car-toward a standardized systematic intercultural agile automotive ui/ux design process," in *Automotive User Interfaces: Creating Interactive Experiences in the Car*, 2017, pp. 143–184. [Online]. Available: https://doi.org/10.1007/978-3-319-49448-7_6

[13] P. Hohl, J. Münch, K. Schneider, and M. Stupperich, "Real-life challenges on agile software product lines in automotive," in *Proc. of PROFES'17*, 2017, pp. 28–36.

[14] International Organization for Standardization, "Software and systems engineering reference model for product line engineering and management," CH-1214 Vernier, Geneva, Switzerland, 01.12.2015.

[15] K. S. Jasmine and R. Vasantha, "A new capability maturity model for reuse based software development process," *International Journal of Engineering and Technology*, vol. 2, no. 1, p. 112, 2010.

[16] F. McCaffery, M. Pikkarainen, and I. Richardson, "Ahaa –agile, hybrid assessment method for automotive, safety critical smes," in *Companion of the 30th ICSE*, 2008, p. 551.

[17] D. Hantke, "An approach for combining spice and scrum in software development projects," in *Software Process Improvement and Capability Determination*, ser. Communications in Computer and Information Science, 2015, vol. 526, pp. 233–238.

[18] K. Schwaber and J. Sutherland, "The scrum guide," 2001.

[19] International Organization for Standardization, "Information technology – process assessment – part 5: An exemplar software life cycle process assessment model," 2012-02.

[20] 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.

[21] S. Weber, "Agile in automotive state of practice 2015," Kornwestheim,, April 2015. [Online]. Available: www.kuglermaag.com/agile2015

[22] A. Schloßer, J. Schnitzler, T. Sentis, and J. Richenhagen, “Agile processes in automotive industry efficiency and quality in software development,” in *16. Internationales Stuttgarter Symposium*, 2016, pp. 489–503.

[23] K. Hayashi, M. Aoyama, and K. Kobata, “A concurrent feedback development method and its application to automotive software development,” in *Proc. od APSEC’15)*,2015, pp. 362–369.

[24] J. Taiber and J. D. McGregor, “Efficient engineering of safety-critical, software-intensive systems,” in *Proc. of ICCVE’14*, 2014, pp. 836–841.

[25] R. Valade, “The big projects always fail: Taking an enterprise agile,” in *Proc. of Agile’08 Conference*, pp. 148–153.

[26] V. Antinyan, M. Staron, W. Meding, P. Osterstrom, E. Wikstrom, J. Wranker, A. Henriksson, and J. Hansson, “Identifying risky areas of software code in agile/lean software development: An industrial experience report,” in *Proc. of CSMR-WCRE’14*, 2014, pp. 154–163.

[27] 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.