

Adoption of Agile Methods in Automotive Software Development

Author: Kenaan Berger
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands

ABSTRACT,

The automotive industry is currently experiencing rapid changes in the external environment. Trends like electro mobility, connected cars, and autonomous driving may have a disruptive impact on a rather static competitive environment. Based on these developments, increasingly changing market demands, for example, the demand for more frequent delivery of innovative product functionalities or safety-related updates, have created a need for shorter software release cycles and more adaptability. However, traditional automotive software development approaches have been found to be inefficient and constrain the release of software in short iterations. Agile software development methods provide a solution for automotive manufacturers to reduce the software release cycle time. Nevertheless, the automotive domain is characterized as a strictly regulated environment, which limits the usage of agile software development methods. This study explores how automotive manufacturers can adopt agile software development methods, under consideration of barriers that have to be overcome. A literature review was conducted, and qualitative data were collected through expert interviews with employees of a German automotive manufacturer, to explore how scholars and practitioners deal with challenges and solutions approaches for the adoption of agile methods in automotive software development. The results indicate that agile methods, due to functional safety standards such as the ISO26262, have to be adapted to the automotive software development environment, but contrary to our expectations, there is not a fixed sequence in which regulatory requirements have to be fulfilled. Next to project management approaches, also organizational cultural and collaboration aspects need be considered, to create an organizational environment that reinforces agility. However, there is not a solution on how to remain efficient, while adhering to regulatory requirements when using the agile method Continuous Integration.

Graduation Committee members:

First Supervisor: dr. M. de Visser

Second Supervisor: dr. A.B.J.M. Wijnhoven

Third Supervisor: MSc L.J.J. Kayser

Keywords

Automotive software development, automotive software engineering, agile methods, scrum, continuous integration, agile transformation

1. INTRODUCTION

1.1 Problem Statement

Nowadays within the automotive industry there are current trends like electro mobility, connected cars, digital transformation, and autonomous driving, which offer new opportunities, as well as the potential for differentiating automotive vehicles. These developments may have a disruptive impact on the established structures of a rather static competitive environment. In line with this expectation, there are new competitors that have increased the competition for market shares. Automobile manufacturers are trying to respond to these challenges, by increasing the competitive differentiation and meeting extended customer requirements through an increasing number of product functions in the vehicles (Fahl et al., 2019). Especially automated driving and connected cars, which are connected to wireless networks through the convergence of automotive and information technologies, are receiving high attention in the automotive industry and academia (Kim et al., 2018; Pfeffer et al., 2019)

Continuous updates will become essential for future automotive systems because vehicles are in us for more than 15 years on average and information technology will improve rapidly during that time span. This means that regulatory authorities will require by law to improve, which could be, for example, to fix possible errors and update crucial components to the state-of-the-art (Oberfell et al., 2019). As a part of these developments, so-called software-over-the-air updates (SOTA) have emerged as an efficient way, in which these, especially safety-related, updates can be released in the future. However, the basis for effective SOTA updates are generally shorter software release and update cycles (Sax et al., 2017).

Increasingly changing software requirements, that come a long with the objective to satisfy market demands, for example more frequent innovation or safety related updates, have created a need for more efficient software development and adaptability. Nevertheless, literature on automotive software development, indicates that current software development processes are too slow to keep pace in future, which forces automotive manufacturers to redesign their vehicle software development processes (Klunder et al., 2018; Agren et al., 2019; Oberfell et al., 2019; Bach et al., 2017).

One of the reasons why the current development processes are inefficient is caused by the characteristics of the V-Model, which is the current state of the art model for the development of automotive software components (Pfeffer et al., 2019). The V-Model can be considered as an extension of the waterfall model, in which the execution of processes happens in a sequential manner (Takahira et al., 2014). As a matter of that, the V-Model is characterized by a strict linear development approach, in which implemented processes follow an iterative procedure on system and subsystem levels and incremental approaches on component and unit levels. This implies that at the beginning of the development project, the requirements have to be detailed and complete. After the development, the software is tested and validated against the specific requirements in a sequential order (Bach et al., 2017). Even though the development organization, caused by the V-Model, hinders information sharing and synergy, the V-Model is implicitly suggested by the ISO 26262 functional safety standard and referenced in the context of the Automotive SPICE standard (Van der Valk et al., 2018; Pfeffer et al., 2019).

Agile software development provides benefits for software developing companies, such as shorter time to market, or stronger customer connections (Diebold and Theobald, 2018). The agile software development models have their origins in the

field of software development. Driven by the fundamental objective of agility, which is to provide customer-relevant results in short iterations, uncertainties or insufficient information, especially during the early stages of the product development process, can result in changing requirements in the later stages of the process. Therefore, the added value from agile methods derives from the ability to react quickly to changing or new requirements (Banijamali et al., 2017). This adaptability to new requirements is enabled by the possibility of incrementally further developing the product and the requirements in short iterations. Nevertheless, agile methods often need to be adopted to a specific context and are in regulated domains, such as automotive, only used to a certain extent or not at all (Diebold and Theobald, 2018). The most used and fastest growing agile framework is Scrum (Ribeiro et al., 2018), and Takahira et al. (2014) conducted one of the first attempts to combine traditional automotive software development processes with Scrum. However, a study by Diebold and Theobald (2018) concludes that the fear of the unknown or lack of understanding of the new processes, as well as the fear of running into regulatory problems and losing certifications, are providing barriers that constrain the adoption of agile methods in the automotive domain.

1.2 Research Project Motivation

As elaborated in the problem statement, the competitive landscape and market demands in the automotive industry are changing rapidly. According to Sax et al. (2017) have different automotive manufacturers used SOTA updates for less safety-critical systems like navigation maps and infotainment applications, but there is currently, worldwide, only one automotive manufacturer that used the SOTA update system for core electronic control units (ECU), which enables for example to update autopilot functions remotely. As general shorter software and update release cycles are the basis for SOTA updates, this provides an example on why automotive manufacturers, especially established ones, are pressured to redesign software development processes, to reduce release cycle times and enable more frequent innovations.

The V-Model, as current state-of-the-art process, assumes that customer and user requirements are complete and correct at the beginning of the development project (Pfeffer et al., 2019; Takahira et al., 2014), and the decomposition of requirements, and many levels of abstractions force early design decisions, which causes delays. Additionally, once the development process has started, it is difficult to adjust changing requirements, since software test cases are also created upfront and are connected to the initial requirements. Next to that issue, the times between the specification of requirements and feedback through integration and acceptance testing, is too long, which then often creates a need for requirements changes at late development stages. A study by Agren et al. (2019) indicates that this decreases the development speed.

Agile methods provide more flexible ways to handle insufficient requirements at the beginning of the development project, by delivering product increments in smaller iterations (Banijamali et al., 2017). As the requirements are updated, before each iteration, and the developed product increment is tested based on the requirements, after each iteration, agile methods enable to continuously refine the final software during the development process and reduce the time to market (Hohl et al., 2018). Also, the issue of late feedbacks can be addressed by agile practices, such as retrospectives, which are conducted after each iteration, to share potential problems like necessary requirement changes (Katumba and Knauss, 2014). Furthermore, researchers state that agile development techniques are a prerequisite for more

continuous development procedures, that allow for innovation in shorter release cycles (Oberfell et al., 2019).

Based on the aforementioned problem statement and the potential added value of agile methods for automotive software development, in respect to the problem situation, my personal motivation for doing this research is to explore ways in which agile methods can be used in automotive software development. Therefore, I decided to analyze current challenges and solution approaches for an adoption of agile methods in automotive software development.

1.3 Research Objectives & Research Question

The objective of this thesis will be to provide an overview of current recommendations on how to enable the effective and efficient usage of agile methods for automotive software development, under consideration of constraining factors. Furthermore, this thesis will explore literature indicated approaches and the approaches of industry experts.

Based on the aforementioned objective, the following research question has been formulated: “How can automotive manufacturers, in light of a demand for more flexible vehicle software development processes, adopt Agile Methods?”

To systematically answer the main research question, two sub questions have been created:

1. What are barriers to the adoption of agile methods in automotive software development, in theory and in practice?
2. What solutions have been developed to overcome the Barriers for an adoption of agile methods in automotive software development, in theory and in practice?

1.4 Outline of this Paper

This paper is basically structured into six chapters. The first chapter includes a theoretical framework, which is supposed provide a short conceptualization of the models that are discussed within the study. The second chapter consists of a literature review, which is based on the two sub questions. The third chapter is a methodology section that explains and justifies the applied data collection and analysis method. Within the fourth and fifth chapters, there will be a more detailed elaboration on how data has been collected and, eventually, the data will be analyzed. The sixth, and last chapter, incorporates a conclusion and aims at synthesizing the theoretical and practical perspectives on the research topic.

2. THEORETICAL FRAMEWORK

Within this chapter there will be an introduction of the theoretical models that are under consideration for this research project, to clarify the scope and conceptualize the concepts that will be discussed. The chapter starts with a description of the traditional automotive software development approach and will be continued by an introduction of agile software development methods in the automotive domain.

2.1 Traditional Automotive Software Development

The traditional V-Model (Figure 1.) is derived from the waterfall model, but emphasizes verification and validation activities within the development processes. In the context of automotive software development, the process starts with the specification of the desired complete vehicle characteristics in a downward movement on the left side of the V. This means there are complete systems requirements down to parts specification,

design, and evaluation steps. After that, the systems, which are developed based on the components specifications, are developed, and tested, as well as validated against their specifications in a hierarchical order on the right side of the V (Liu et al., 2016). Agren et al. (2019) state that there are many safety and legal concerns in automotive software development, which makes the decomposition of requirements and testing necessary, nevertheless can the usual way, to do it upfront development, cause preventable delays.

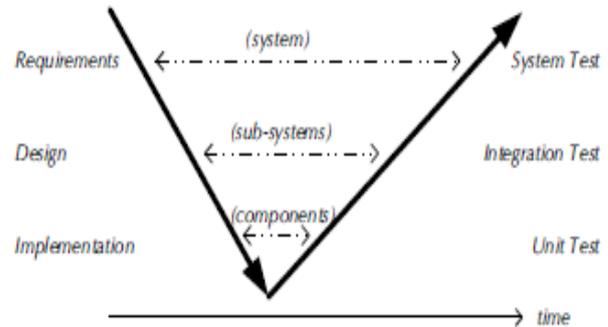


Figure 1. Traditional V-Model (Liu et al., 2016)

2.2 Agile Automotive Software Development

There are many different agile methods, which applicable in different contexts. In general, the main agile methodologies that were discussed in literature are Crystal methodologies, Dynamic Software Development Method (DSDM), Feature-Driven Development, Lean Software Development, Scrum, and Extreme Programming (XP) (Dybå and Dingsøy, 2008). Diebold and Theobald (2018) state that there has been one study by the German consultancy firm Kugler Maag Cie, which covered, contrasting to other studies that did not explicitly cover embedded domains, agile in automotive. The respondents of the study indicated that the agile methods Continuous Integration, Test Driven Development, Feature Driven Development, Extreme Programming, Scrumban, Kanban and Scrum were all used in an automotive context. Among the respondents of the study, the following agile methods were used most often in an automotive software development context: Scrum with 79%, Continuous Integration with 64% and Kanban with 55%. Furthermore, the respondents explain that so called hybrid models were used, which combine agile methods with traditional software development processes (Kulger Maag Cie, 2015).

Heerwagen (2018) explains that Kanban, unlike Scrum, is more suitable as a facilitator for debugging and maintenance. However, the research question focuses on software development processes, so this section will solely introduce the concepts of Scrum, Continuous Integration, and Hybrid Models, to clarify the scope of this research.

2.2.1 Scrum

Out of the numerous agile frameworks, Scrum (Figure 2.) is one of the most popular and fastest-growing frameworks, which is used all over the world by Information Technologies (IT) companies. A Scrum team consists of a Scrum Master, a Product Owner, and a Development Team. The Scrum Master is responsible for supporting and guiding the team towards a successful project and use of Scrum. The Product Owner gathers all the information on what should be produced from customers or end-users, as well as other stakeholders, and translates this into a prioritized list, to assure the team is working towards the desired end goal. The Development Team is responsible for the development of the product, under consideration of the Product Owner's requests (Takahira et al., 2014).

Scrum organizes product development in incremental cycles of work, called sprints. Sprints are time-boxed typically with the same duration and are driven by the objective to deliver a product increment. As soon as a sprint has come to an end, a new sprint starts. The sprint cycle starts with a stakeholder meeting in which the project vision is agreed on and the Product Backlog is created and prioritized by the Product Owner. The Product Backlog is a list of requirements that is prioritized by business value. The sprint cycles begin with a sprint planning meeting to decide which requirements of the Product Backlog will be included in the sprint. During the sprints, there are daily stand-up meetings with 15-min maximum duration, to synchronize the team members. In these meetings the team members share what they have done so far and if they are facing difficulties with something. Towards the end of a sprint, stakeholders and the

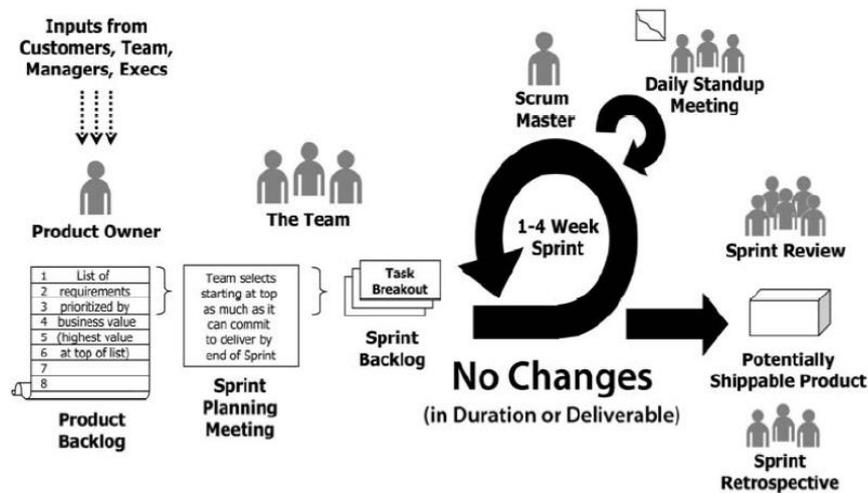


Figure 2. Scrum Software Development Process Phases (Takahira, 2014)

development team conduct a sprint review meeting to validate the increment that has been delivered and assure the Product Owner approves it. The last step of a sprint is the sprint Retrospective Meeting, which provides an opportunity to the Scrum team to reflect and reevaluate their performance (Ribeiro et al., 2018).

2.2.2 Continuous Integration

As one of the Extreme Programming practices Continuous Integration has become popular in software development (Beck, 1999). Even though there is no consensus on continuous integration as a single, homogeneous practice, the shared understanding of Continuous Integration implies that software developers integrate code frequently into a central repository at least daily (Stahl and Bosch, 2014). According to Goodman and Elbaz (2008) does Continuous Integration improve release frequency and time to market.

Obergfell et al. (2019) elaborate that Continuous Integration helps to improve the quality and speed at which automotive software-based innovations are delivered to the customers. They also point out that in regard of autonomous driving, for example an encryption algorithm used for backend communication can become outdated and is not secure anymore, which requires the continuous mitigation of attack surfaces. Marner et al. (2019) state that the incremental integration of smaller work products can replace a larger and more complex final integration and helps to achieve transparency about the finished content of the release as well raising awareness of dependencies. Furthermore, Kassner et al. (2016) explain that Service oriented architecture, which is needed for continuous improvement, can enable the usage of real time qualitative and quantitative data for product software improvement.

2.2.3 Hybrid Models

Kuhrmann et al. (2019) have defined Hybrid models as “...any combination of agile or traditional (plan-driven) approaches that an organizational unit adopts and customizes to its own context needs.”

As the automotive domain is dealing with safety-critical software, it is strictly regulated. In line with this a common practice for software development is the aforementioned V-Model. This situation has led to the creation of hybrid models that focus on speeding up the development phase and implement the best principles of agile methods like Scrum inside the V-Model methodology to assure the required level of quality and safety (Takahira et al. 2014).

As Automotive Software Performance Improvement and

Capability dEtermination (ASPICE) is a major software process development standard for car manufacturers, it builds the basis on which automotive software development processes are assessed, in terms of software and process quality (Diebold et al., 2017). In line with this, there are so called traceability matrices which show to what extent the best practices of automotive SPICE are covered within hybrid models (Hantke, 2015).

3. LITERATURE REVIEW

To answer the two sub questions from a literature-based perspective, this chapter will encompass challenges and solution approaches for an adoption of agile methods in automotive software development, which are currently discussed in scientific literature.

3.1 Challenges for an Adoption of Agile Methods in Automotive Software Development

The aim of this section is to provide insights on the literature about domain specific challenges, which potentially constrain the adoption of agile methods for automotive software development. Since different agile methods come with different challenges, this section is divided into subchapters that explore challenges for Scrum and challenges for Continuous Integration.

3.1.1 Challenges for Scrum in Automotive Software Development

The literature review on challenges for the adoption of Scrum in automotive software development points out several smaller barriers that can be allocated to three main challenges. The identified main challenges are safety-related challenges,

organizational cultural challenges, and knowledge-related challenges.

Safety-related challenges result from the high criticality level of in-vehicle software, of which some of them are life-critical. To assure the high level of quality for reliability, functionality, efficiency and other characteristics, the quality assessment standard Automotive SPICE (A-SPICE) is commonly used by automotive manufacturers (Komiya et al., 2019). In line with this, Komiya et al. (2019) recognized multiple barriers when applying A-SPICE to Scrum. Among these barriers were the assurance of a correspondence between agile process and A-SPICE software engineering processes, a lack of documentation during development, and the establishment and maintenance of traceability among work products, which can flexibly respond to changes. Marner et al. (2019) state that in regard of Scrum this means that, on the one hand, legal requirements, standards, and production requirements must be taken into account and, on the other hand, arrangements with software developers, who urge to act more freely without being restricted by guidelines, has to be made. Additionally, Steghöfer et al. (2019) point out that scaled agile development of safety-critical systems is dependent on traceability and the ability to prove regulatory compliance at any point in the development process.

Sandu and Salceanu (2019) explain that knowledge-related deficiencies can have severe consequences, as a crucial success determinant is that the working system, which will be delivered after each sprint, does not have major malfunctions and the end-user functionality should not be affected. Thus, Scrum development teams need to have effective and efficient testing processes, as well as defect detection mechanisms. Nevertheless, they state that real projects implementation showed that insufficient experience of developing manual and automated test cases results in escaping defects. Those defects are discovered afterwards by the system team, but with a delay of potentially several iterations until the complete system is integrated and tested. The new discovered critical bugs then need to be fixed as soon as possible, which affects the planning for the following iterations, so that a domino effect can be created that may affect the objective of the entire program as well.

According to Heerwagen (2018), organizational cultural challenges derive from the issue that when a company seriously implements agile development methods, it can lead to the obsolescence of certain jobs, which therefore affects the level of enthusiasm. For example, the team leader layer is not necessarily needed anymore because, depending on the task, there is only a Scrum Master and a Product Owner alongside a small development team. Furthermore, a study conducted at the Porsche AG, has shown that currently decisions are still made at higher levels of hierarchy and the development team does not have enough decision power. This is contrary to the idea of Scrum, since the development team is supposed to know best, what decisions are appropriate (Marner et al., 2019).

3.1.2 Challenges for Continuous Integration in Automotive Software Development

The literature review on challenges for the usage of Continuous Integration in automotive software development emphasizes two main challenges that consist of multiple smaller barriers, which can be allocated to the two main challenges. Factors that challenge the adoption of Continuous Integration are safety-related challenges and challenges with cross-collaboration.

Gaiimo et al. (2019) explain that the automotive domain is a context, in which it is more difficult to roll out changes and new software as quickly as possible since safety is fundamental. Furthermore, regulations impose restrictions on what hardware and software are allowed on public roads, which has resulted in

a need for lengthy testing and verification processes before a software code can be changed. In line with this issue, Schlichthärle et al. (2020) recognize that in a large-scale agile software development project, there need to be mechanisms, which always ensure a certain level of functional quality, even if developers are autonomously integrating and changing software code. Additionally, Vost and Wagner (2017) emphasize the need for updates of the safety analysis with every change that is made.

Collaboration challenges encompass different barriers inside and across organizational boundaries. Van der Valk et al. (2018) explain that due to the V-model, which is implicitly suggested by the ISO 26262 functional safety standard, the overall development organization is split into different groups. These different groups, for example divided into different competences, can cause a silo effect that hinders information sharing and synergy. Across organizational boundaries, collaboration challenges derive from the issue that many stakeholders must be considered, including suppliers, since continuous deployment depends on their contributions, as they produce certain electrical and software components that are later integrated by the automotive manufacturer (Pellicione et al., 2017; Van der Valk et al., 2018). Another constraining factor in this context is that current collaboration models between automotive manufacturers and suppliers are commonly based on written specification documents (Oberfell et al., 2019). However, even though strict contract-based collaboration is constraining inter-organizational Continuous Integration, contracts facilitate negotiations between different organizations (Van der Valk et al., 2018).

3.2 Solutions for an Adoption of Agile Methods in Automotive Software Development

Similar like the section for the challenges, this section will encompass current solution approaches that have been developed to adopt Scrum and Continuous Integration in automotive software development.

3.2.1 Solutions for Scrum in Automotive Software Development

As A-SPICE is the most popular quality assessment framework for automotive software development, Komiya et al. (2019) have defined hybrid processes incorporating both agile and waterfall aspects, which means certain aspects are defined by Scrum and other aspects by automotive SPICE. The goal of this approach is to benefit as much as possible from the increased flexibility and increased development speed of Scrum, but at the same time to assure that legal requirements are met.

Figure 3. provides the process overview of the hybrid process approach. While the preparation and release sprints are defined using waterfall processes, to ensure a high level of quality, the development sprints are defined on a Scrum basis. This means the creation of the software requirements and the architecture design are conducted within the preparation sprint. Afterwards a flexible development sprint starts, which is based on Scrum and contains first unit and functional tests. If the retrospective meeting is satisfactory, the release sprint starts, and the software is tested based on testing requirements that are recommended by A-SPICE. Also, the issue of insufficient documentation is solved since the creation of minimum required documents starts with the development sprint and is completed within the release sprint.

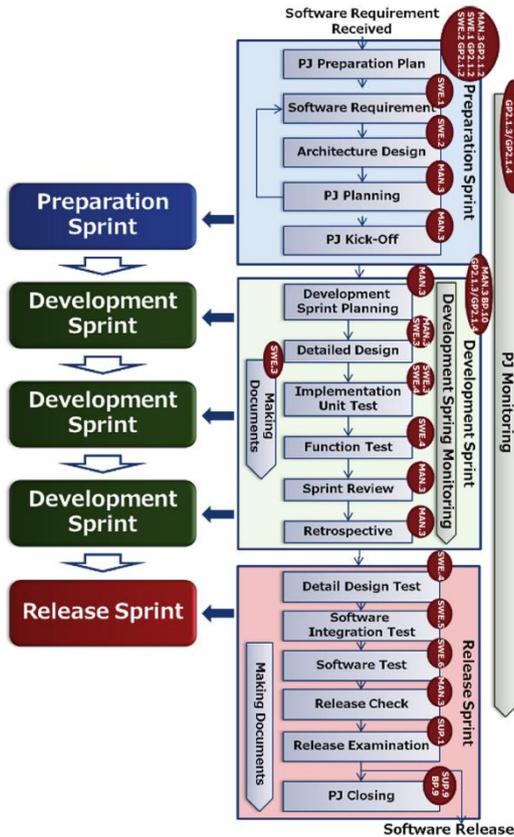


Figure 3. Overview potential Hybrid Processes (Komiya et al., 2019)

Steghöfer et al. (2019) propose possible solutions for the traceability and compliance barriers that arise when applying Scrum in large scale, safety-critical projects. The proposed solution includes the establishment of a traceability information model (TIM), which supports the safety analysis and the creation of continuous compliance mechanisms, to keep safety-related artefacts, such as safety cases that provide evidence that a specific functionality is acceptably safe, up-to-date. Steghöfer et al. state that a toolchain that supports traceability, could help to identify if a safety case needs to be changed and integrate variants into the handling of safety arguments. Furthermore, they emphasize that continuous compliance could be enabled by techniques that allow the incremental update of the safety case based on individual change requests to achieve a reduction of

cost and time required for changes and enable the integration of supplier components in the system.

Marnier et al. (2019) provide a way to approach the challenge of additional qualification phases to fix remaining bugs or to finish functionalities that had been planned for the previous sprint, and therefore cause that planned results for the next sprint cannot be fully achieved. The approach that is suggested by Marnier et al, focuses on a definition of done (DoD) and the incorporation of time-boxed sprints, to increase transparency regarding the content that was finished within an iteration. Another aspect of this approach is an assessment of the status quo at the end of each sprint, and the creation of a planning for unfinished requirements for the next sprint. Additionally, the creation of transparency also involves the conduction of retrospectives together with relevant stakeholders and dependent projects. Regarding the issue of the creation of adequate test cases, Sandu and Salceanu (2019) propose to enable knowledge sharing by exchanging members of the system teams with members of the Scrum teams, so that they can teach their colleagues how to write meaningful manual and automated test cases on team level and how to automate and execute the regression tests. The objective of this approach is that each agile team member will have the competency and necessary skills for system testing as well.

Furthermore, multiple studies suggest adopting agile methods, like Scrum, with an incremental, stepwise approach rather than a big bang approach (Hagihghatka, 2017; Diebold and Theobald, 2018; Komiya, 2019). Diebold and Theobald (2018) elaborate that a stepwise approach helps to build acceptance, but another crucial component is a common goal for a team, which has been found to enhance the team moral, productivity and motivation.

3.2.2 Solutions for Continuous Integration in Automotive Software Development

According to Schlichthärle et al. (2020) an essential success factor for Continuous Integration, in a large-scale agile software development projects, is to possess a master repository, in which a stable basis of quality is ensured. To maintain a high level of software quality, they suggest avoiding that people commit changes into the master repository without any quality check. Based on this suggestion, they explained the process model of a staged development process, with semi-automated quality gates (Figure 4.). The process starts with a developer that creates a small, short living, development branch and does the changes in the branches. After that the developer initiates a pull request to take over the changes in the master repository. Once the pull request has been triggered, two stages of automated quality gates are triggered, which are a check and a gate. In order to pass the check, at minimum one manual review by another developer has to be conducted. After that, the change is either declined with

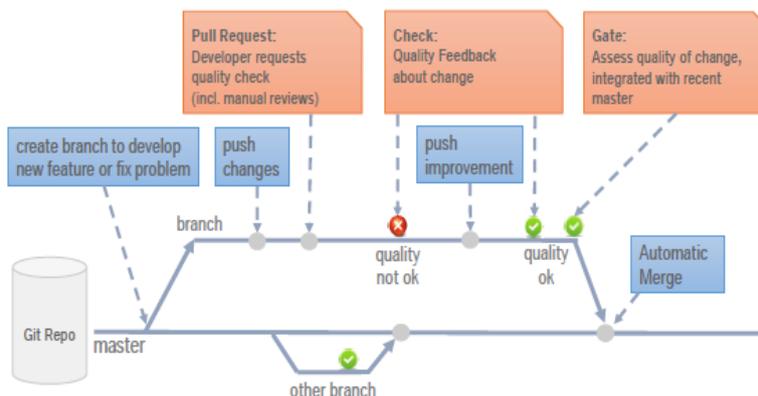


Figure 4. Continuous Integration process with Quality Gates (Schlichthärle et al., 2020)

comments that request an improvement, or the change is approved. When the review and all automated tests, which are performed within the check, are acceptable, the gate is triggered and performs additional, more detailed, tests in a virtual merge with the master repository. Finally, if the automated tests of the gate are also acceptable, the change will be merged into the master repository.

Vost and Wagner (2017) suggest multiple guidelines to keep continuous delivery safe. The first guideline advises to perform iterative safety analysis parallel to development, as an up-to-date safety analysis is considered the first prerequisite for continuous safety testing. The objective of the second guideline is to reduce the manual work, in regard of safety case generation, to a minimum, by implementing automated safety test execution and generation. The third and fourth guideline require to handle safety analysis and its results as every other artifact that is required for a build, which means that the analysis must be stored in a central repository in a versioned manner. In that way every change in the analysis can be tracked. Furthermore, there has to be a safety test in every build, to ensure that every build that might be delivered to production has passed a safety check, so that potentially unsafe software is not deployed.

Van der Valk et al. (2018) conclude that even though strict contracts are providing stability and leverage to the parties, more flexible contracts, so called agile contracts, are needed to improve the supplier collaboration. Furthermore, studies indicate that transparency, which is, in this case, the degree of information that is shared among the organizations operating in the same value-chain, is an enabler for inter-organizational Continuous Integration (Van der Valk, 2018; Pellicione et al., 2017). Additionally, Obergfell et al. (2019) identify IT-Architectural drivers like, IP-based architecture, extended electric architecture and learning architecture, as enabling technologies for collaboration and development in automotive software development projects that use Continuous Integration.

Overview of literature-based Challenges for Scrum in Automotive Software Development	Overview of literature-based Solutions that address the Challenges
<p><u>Safety:</u></p> <ul style="list-style-type: none"> - High criticality of in-vehicle software - Functional safety standard ISO26262 demands strict safety-requirements - Mismatches between the quality assessment model A-SPICE and Scrum - Tradeoffs between legal compliance and software developer autonomy 	<ul style="list-style-type: none"> - Implementation of hybrid processes that exploit the benefits of Scrum but ensure regulatory compliance - Establishment of traceability information models to support safety analysis and continuous safety mechanisms - Techniques that allow an incremental update of safety cases to enable continuous regulatory compliance
<p><u>Knowledge/Expertise:</u></p> <ul style="list-style-type: none"> - Late discovery of defects, which escaped an iteration, by the system team - Insufficient knowledge of the Scrum development team on how to develop effective test cases - Difficulties to ensure timeliness and provision of a product increment, without major malfunctions at the end of an iteration 	<ul style="list-style-type: none"> - Creation of a well-defined DoD - Incorporation of time-boxed sprints and an assessment of the status quo at the end of each sprint - Conduction of retrospectives together with relevant stakeholders and dependent projects to increase transparency regarding the content that was finished in an iteration - Knowledge sharing by exchanging members of the system team with members of the Scrum team
<p><u>Organizational culture:</u></p> <ul style="list-style-type: none"> - Traditional environment with a resistant to changes - Job profiles and responsibilities need to be adapted to the Scrum framework - Decision authority is still secured within higher hierarchy levels 	<ul style="list-style-type: none"> - Adoption of Scrum with a stepwise approach to build organizational acceptance - Formulation of common goals for a team to enhance team moral and motivation

Table 1. Challenges and Solutions for an Adoption of Scrum in Automotive Software Development based on Theory

Overview of literature-based Challenges for Continuous Integration in Automotive Software Development	Overview of literature-based Solutions that address the specific Challenges
<p><u>Safety:</u></p> <ul style="list-style-type: none"> - Difficulty of committing software changes quickly, as safety is fundamental in automotive software development - Strict regulations on what hardware and software is allowed on public roads - In large-scale agile software development projects, a certain level of functional quality has to be ensured at any time - Every software change demands an update of the safety analysis 	<ul style="list-style-type: none"> - Usage of a master repository as a stable basis - Quality gates that prevent that people commit changes to the master repository without quality checks - Iterative safety analysis parallel to development - Reduction of manual work and implementation of automated safety test execution and generation - Enabling of change tracking, by storing the safety analysis and its results in a central repository
<p><u>Collaboration:</u></p> <ul style="list-style-type: none"> - Silo effect, caused by the ISO 26262 functional safety standard, hinders information sharing and synergy - Many stakeholders, such as suppliers, need to be considered to enable continuous deployment - Currently there is strict contract-based collaboration, which constrains inter-organizational Continuous Integration 	<ul style="list-style-type: none"> - More information sharing among organizations that operate in the same value chain - Introduction of more flexible contracts, so called agile contracts - IT-Architectural drivers as enabling technologies for collaboration

Table 2. Challenges and Solutions for an Adoption of Continuous Integration in Automotive Software Development based on Theory

4. RESEARCH METHODS

The aim of this research project is to provide an overview of the current challenges and solution approaches for an adoption of agile methods in automotive software development. The purpose behind this research objective is that current trends like electro mobility, connected cars and autonomous driving, require flexible development processes and the ability to react quickly to changing requirements. Researchers and industry experts state that the adoption of agile software development methods provides a way to address these changes in market demand. Nevertheless, strict regulations and safety critical software components, which characterize the software development landscape in the automotive domain, provide barriers to the agile way of software development.

At the start of this research, desk research about agile methods in general and automotive software development has been

conducted, to obtain an understanding of the underlying concepts. The sources for this information were the databases Scopus and Web of Science.

To systematically answer the research question, it was decided to use theoretical knowledge that has been gathered from scientific literature, and a non-comparative analysis of qualitative data, which has been collected through expert interviews with employees of a German automotive manufacturer.

The reason why a non-comparative analysis of qualitative data has been chosen for the research design is that the processes and accompanying aspects of automotive software development are complex, and the relationships of different aspects have to be understood. Furthermore, the adoption of agile methods can be addressed from multiple perspectives, which means there are also multiple ways to answer the research question. Therefore, the objective of this study cannot be reached by searching for one best answer, but by exploring different ways to address the topic, which means that data collection had to be conducted in an exploratory context. In line with this, quantitative data, like counts and measures, have been excluded.

The literature review and the results of the interviews, as well as the qualitative data analysis will be done in Chapters 5 and 6 and are going to lead to the conclusion of this research paper and to the answer of the main research question.

4.1.1 Literature Review

The main search engines to find relevant literature for the literature review about the challenges and solution approaches for an adoption of agile methods in an automotive software development context were, Web of Science, Scopus, and Google Scholar, as well as Google. Since Scopus and Web of Science are databases that provide peer-reviewed literature and quality web sources, they were favored over Google Scholar and Google. In Scopus and Web of Science, it is possible to use keyword search and, in line with the research topic, the following search strings were used: “Agile” AND “Automotive”, or “Agile” AND “Automotive Software”, or “Agile” AND “Vehicle”, or “Agile” AND “Vehicle Software”. Furthermore, the search strings were continued by replacing “Agile” with “Scrum” and “Continuous Integration”. As Computer Science tends to be a domain in which changes and developments occur frequently and the goal of this research is to provide an added value to automotive manufacturers, it was decided to solely focus on current challenges and solution approaches. To ensure actuality of the challenges and solution approaches, the results of each keyword search were limited to the years 2017-2020. The review process of new papers was stopped, when there was increasingly repetitive information and no new contributions anymore, that were considered to be useful for the scope of this project.

4.1.2 Expert Interviews

To find competent interview partners, it was decided to include agile software development experts from different levels, under the condition that they have actively participated in an agile software development project for automotive functionalities. As this project is conducted in an explorative context, it was important to include people from different hierarchical levels, to explore the research questions from different perspectives. In line with this, five experts from a German automotive manufacturer participated in structured interviews to collect qualitative data. Table X provides an overview of the experts and their roles. The interviewees are called E, followed by a specific number.

Expert	Role
E1	Scrum Master
E2	Department Leader/Manager
E3	Technical Leader/ Product Owner
E4	Manager/Product Owner
E5	Senior Software Developer

Table 3. Selection of Expert Interview Partners

The interviews are started with a fixed set of questions, but this set will be divided into three categories. The first set asks general questions asked about the biggest challenge that had be dealt with when using Scrum, and general experiences with Continuous Integration in automotive software development, to obtain an impression of which challenges the experts consider as the most important barriers for an adoption of agile methods. The second set included two, more specific, questions about safety assurance mechanisms when adopting Scrum and Continuous Integration. The goal of the second set is to explore, if there are complementary, as well as validated solutions to the existing approaches, which are elaborated in the literature review. The last question asks about the expert’s opinions on the claim that Scrum cannot be used by the book but hast to be adapted to the automotive software development context. The aim of this question was to investigate whether Hybrid Models are actually used in the automotive industry and if the approaches are different than the ones that were analyzed during the literature review. Appendix 12.1 shows the questions that were asked during the interviews.

5. RESULTS

Within this chapter, the results of the interviews will be presented. To provide an overview of the results in the context of the research question, this chapter is structured around the two sub questions.

5.1 What are barriers to the adoption of agile methods in automotive software development? (sub question 1)

With respect to sub question 1, it appears that the experts have different opinions on what the most challenging factors, in regard of Scrum, are. Also, for the usage of Continuous Integration, some of the experts have encountered more severe difficulties than others.

5.1.1 Constrained agility through fixed release planning

It was explained that Automotive software development projects have fixed end dates, which means it is not possible to flexibly extend the development processes whenever there are uncertainties or improved understandings of underlying issues. At the end of the planned development period the product, in this case the car, has to be finished, which provides a conflict between Scrum and traditional automotive development.

- *“I would say the most challenging aspect of Scrum is the plannability so to say, so in the automotive industry the deadlines are strict and the plans are there, and forecast are important, and also to be sure if you can meet them or not. And when applying agile methodology, it comes to a point where you have uncertainty in your doings because you learn parts of the work while doing it. Uncertainty is a key issue when dealing with Scrum because it is not as easy to show*

confidence in reaching your goals, to upper management partners etc.” - E2

Furthermore, current development methodologies, specifically the V-Model, are creating more rigid requirement engineering processes that are challenging the ability to react to changes within Scrum.

- *“I think the biggest challenge of Scrum in automotive is, to run Scrum or agile methods in the V-Model process, since the V-Model is still needed and mandatory in automotive software development. The main issue with that is having a fixed set of features, which are very detailed and even defined a few years before the delivery, as well as the due date of delivering all the features with at least half a year before the final product is on the market. Then there is half year just for bug fixing saved, with in theory, no possibility to adapt features to what has emerged new on the market.” - E4*

5.1.2 Hardware dependencies and complex safety requirements

In regard of the usage of Continuous Integration in automotive software development, the experts explained that hardware dependencies are reducing the development speed.

- *“I think Continuous Integration will always be there, but in automotive it is difficult because it is embedded, and embedded is always limited by hardware resources you have. The simulations we do have, are not always sufficient. The actual Continuous Integration need to be optimized in the case of automotive, especially in projects as big as ours, it is sometimes challenging. Since we cannot get everything into the CI, we would like to but since we cannot, we sometimes find out issues too late.” - E3*

Furthermore, one expert states that strict safety-related requirements, combined with hardware dependencies, are currently making Continuous Integration inefficient in automotive software development.

- *“My experience with Continuous Integration in automotive is not the greatest because of the number of tests we have to do because of ISO. So here automotive is really different to what I have experienced before. To put it into perspective, in automotive we have this ISO 26262, we have all this stuff about like norming, and its very constraining because you need to test the software as a whole. The problem is the ISO has inflated to a point which is ridiculous. No one, not even my boss, who is an ISO person expert, like legal person of ISO, can remember all aspects he needs for his small part. Continuous Integration reflects this, right now our CI, if I try to commit something right now, even a comment on the nightly batch will take about 9 hours, 1 hour for the check pipeline, we have to find the code owners and fight to get their attention, get it validated, get through the gate for two or three hours. There are so many tests that some of them are massively inflated, so that we have to redo them and maybe have to pass them two or three times, so at the end of the day I will pass my comment. At the same time, a big part of your code is code that is extremely hardware dependent. For*

example, driver detection, signal interpretation etc., part of it needs to start on the computer in the CI. You cannot, so there is a lot of code that is not really coverable by unit tests, but you are required to do branch tests anyways. I mean, it's a headache!" - E5

5.1.3 Challenges with organizational cultures

There are multiple experts that put emphasis on the importance of an organizational culture that reinforces agility and the adoption of agile methods. Especially the relationship between the development team and the management has been criticized.

- *"In general, my problem with Scrum and Kanban and all that stuff of the agile methodology is, that it is really far from the agile manifesto origin and it is very infantilizing. It is often used by Managers as a mean of controlling, for example they are always trying to make sure you do everything. As a young developer I did appreciate that it gave me a structure, as I had an architecture that came from a senior developer, so I could trust the architecture that I was given and I had a plan on what to do. As a more senior developer it is kind of insulting to be treated that way and it is also very frustrating in terms of motivation. I feel like Scrum in general works well with young, a little more junior developers, but if you have more experienced developers, then it becomes very soon very tense between manager and developer."* - E5

Next to the issue of trust, it was also pointed out that motivation, as well as the right mindset are crucial aspects that are challenging existing corporate cultures in the traditional automotive domain.

- *"If all the people are motivated, if you know the agile manifesto, one of these principles is exactly about "build the projects around motivated individuals" provide them with the right environment and trust them to get the job done, this is exactly the point. If you cannot assure that people are motivated, not just work, trying to code and test etc., but exactly achieve the common goal, it will not give you real value."* - E1

5.2 What solutions have been developed to overcome the Barriers for an adoption of agile methods in automotive software development? (sub question 2)

Regarding sub question 2, the experts suggest, opposed to the challenges, similar solutions for safety and quality related barriers, as well as organizational cultural and collaboration challenges of Scrum and Continuous Integration in automotive software development. Nevertheless, there are different recommendations for the barriers that were individually perceived as most challenging.

5.2.1 Agile collaboration and organizational culture

Regarding the barrier of fixed released dates, which build up pressure on the Scrum teams, it was proposed that inter-organizational openness, especially towards customers, regarding challenges and progress are essential.

- *"And the way to deal with that is maximum transparency in my perspective, so you have to be very*

transparent on your goals and on your working process and always keep customers up to date and inform them how far your projects are and communicate more I would say." - E2

One aspect of transparency is also to include important stakeholders, like the customers, within the development process.

- *"Well, we basically need to update the requirements during the development, so we need to eventually convince even the customers or project leaders that some changes need to be done. Important is that they are up to date and we can deliver the best product possible"* - E4

Additionally, it was also explained that good communication and collaboration, internally, is needed, to avoid substantial changes at the end of the development project which could potentially cause delays. This means that if multiple teams work on small product increments, it is important that there is constant alignment of the team progress.

- *"We create small meetings or we are just advising each other on the things we do, so that the bigger teams can see what is already there and we always call in for features so if they need something, we just tell them to come forward and toss their use case, we can plan it accordingly. In this way, we can avoid these sudden changes. We are just trying to be more open about it."* - E3

Next to collaboration approaches, some of the experts also point that agile should not solely be viewed as a process, but more as way of thinking. In line with this, trust was stated to play a crucial role.

- *"I often say that you cannot apply any process if you don't adapt the company culture, it is just another process, it's just another paperwork. In most cases if an agile process is introduced, you just get another process for controlling developers. I think the hardest part of Scrum is too really get people to relax and the management to actually trust the people."* - E5

Furthermore, it was explained that Scrum has to be continuously adapted and should also remain flexible in the way it is applied, which means that rules are subject to change.

- *"In my perspective agile is a mindset, not only writing down a set of rules, it is about understanding the principles. Of course, rules are important, but you have to check on the rules from time to time and see which of them are still valid and in which form. It should be viewed as a living process I would say."* - E4

5.2.2 Safety as an ongoing set of activities

In regard of safety related requirements, like the quality assessment framework automotive SPICE, the experts are agreeing that detailed requirements and testing are absolutely necessary but there is not a sequential order in which these activities have to be done. In the end the most important factor is that the requirements have been fulfilled and the tests have been completed and passed.

- *“The safety requirements do not demand you to use certain models, only that they are met in the end. This means the “how” can be variable.”* - E1

It was explained that the requirements are created incrementally and that for every requirement in Scrum, it is necessary to create a test case, so that in the end of each sprint it has to be ensured that the created software increment adheres to that test case and the fulfillment of the requirement is given. One expert explains that the Scrum element Definition of Done (DoD), which has to be constantly updated, ensures that these objectives can be met.

- *“Scrum is designed to ensure high quality, there is even a specific element called DoD or definition of done. Imagine that a team starts with something and then they face this huge amount of requirements for safety, it actually doesn’t matter what the requirements are for because they put the requirements into the DoD and then they start complying with these requirements from the very beginning. It is very important to have a clear vision on how your Definition of Done must improve over time.”* – E1

In regard of safety mechanisms for Continuous Integration, all of the experts state, that there is a system that performs built in tests and provides the opportunity to monitor. Furthermore, there are test cases or even matrices for functionalities, and a gating process that only allows developers to commit certain features if they are compliant with the matrix and tests.

- *“So the answer is gating, strict test automation and a very rich set of code reviews, not letting everybody just commit changes, but having a voting and gating for all the commits happening to the system and having a strict notion of few requirements in terms of KPIs and test coverage. So by via a pull request it possible to change only a specific part.”* – E2

5.2.3 Hybrid Models

All of the experts agree that Scrum and agile methods in general have to be adapted to the automotive software development context. Even though the order of process steps is subject to variance, it was stated that Scrum and the V-Model are synthesized.

- *“Scrum is an iterative application of the V-Model.”* – E3

As automotive software development is characterized by hardware dependencies, certain development processes on the hardware still have to follow waterfall processes, whereas purely software related steps can be conducted using agile methods.

- *“On the hardware side certain processes cannot be agile since you have fixed dates etc. That is the reason why you need the V-Model but on the lower level, or on the more detailed level we can really apply Scrum. In my perspective, the best way is to make a little mix and I think the normal Scrum is never really applicable in practice.”* – E4

6. DATA ANALYSIS

Within this chapter, an analysis of the results, which were presented in chapter 5, will be conducted. The focus of the analysis will be to investigate, which challenges and solutions for an adoption of agile methods in automotive software development that have been discovered in theory, were validated, and which additional challenges and solutions for an adoption of

agile methods in automotive software development were mentioned by the interviewed experts.

6.1 Validated Challenges and Solutions for an Adoption of Agile Methods in Automotive Software Development

6.1.1 Validated challenges

It was stated by one expert that it is challenging to run Scrum within the V-Model, which is still mandatory and creates rigid design requirements engineering processes. Therefore, it was validated that regulatory standards and strict design requirements are currently constraining the flexibility of Scrum in automotive software development.

Furthermore, it was validated that complex safety requirements have to be incorporated into Scrum and Continuous Integration processes to ensure regulatory compliance and manage safety risks. There was also an expert that complained about inflated testing and verification procedures that have to be followed to commit software changes when working with Continuous Integration. That means, that also the difficulty of rolling out changes and new software quickly due to lengthy testing procedures, has been validated.

Also, for organizational cultural challenges, the experts’ perceptions seem to be congruent with the literature for the most part. Especially, in regard of the necessary decentralization of decision authority, there are validated challenges, for example one expert criticizes that often times the development team is not allowed to work completely autonomously because managers, contrary to the agile idea, still want to control the development process.

6.1.2 Validated solutions

All of the experts agree that Scrum has to be adapted and synthesized with certain waterfall processes, such as elements of the V-Model, this validates the suggested usage of hybrid processes that were also mentioned and discussed in the literature.

In terms of necessary quality assurance, to fulfill safety requirements within the Scrum sprints, one expert explains that the so-called definition of done (DoD), provides the means to ensure requirements compliance right from the beginning of the development process. This solution has also been proposed by the literature. In line with the DoD concept, literature and experts agree that it is essential to be transparent about the content that was finished, as well as difficulties that were encountered within a sprint, and communicate this to important stakeholders, such as customers.

Furthermore, an expert validates the assumption that it is also important to establish team goals that are transparent, and that enable the Scrum team members to identify themselves with. This was stated to be important to fulfill one of the principles of the agile manifesto, which is, to build projects around motivated individuals, since a common goal is said to increase the motivation of the individual team members.

Equal to the literature, the experts explain that for a Continuous Integration process that is regulatory complaint and assures quality, it is necessary to have automated tests and quality gates that have to be passed in order to commit changes. It was also validated that a master repository is used to provide a stable basis for quality and being able to provide evidence for regulatory compliance at any time during the development process.

6.2 Additional Challenges and Solutions for an Adoption of Agile Methods in Automotive Software Development

6.2.1 Additional challenges

Complementary, to the literature-based challenges, one of the experts explains that plannability and strict deadlines of forecasts are necessary, but are constrained by uncertainty, which is omnipresent when working with Scrum in automotive software development. This makes it difficult to confidently communicate to important stakeholders whether forecasts and goals can be met, even though the release dates are fixed.

There is also an additional challenge for the usage of Continuous Integration in automotive software development that was mentioned by multiple experts. Automotive software development takes place in an embedded systems environment, which means there are hardware dependencies. However, the experts state that it is not always possible to synchronize hardware and software development, and, sometimes, issues are discovered too late, which then causes further delays.

6.2.2 Additional solutions

In respect to quality and safety assurance, when using Scrum in automotive software development, the experts explain additionally, that the V-Model is mandatory, but Scrum is used at the bottom part of the V-Model, which means that the Scrum teams are responsible for the creation of the software requirements as well as for the conduction of the related testing. The difference between the approach that was explained by the experts, and the approach that was stated within the literature, is based on the literature-based assumption that the V-Model demands to conduct requirements design and testing procedures in strict sequential orders, but multiple experts point out that the

order can vary and the most important factor is that the requirements and tests are appropriate and have been fulfilled. Furthermore, the DoD is responsible to ensure that the product increment that was produced within a sprint, fulfills the necessary requirements.

Additional solution approaches were found in form of recommendations on the appropriate organizational culture that reinforces the usage of agile methods. Next to the issue of decision authority and trust, which was also discussed within the literature, multiple experts also point out that agile should not solely be seen as a process, but rather as an organizational mindset. Part of this organizational mindset is that rules are not fixed and should be continuously adapted and renewed.

Validated challenges and solutions for an adoption of agile methods in automotive software development	Additional challenges and solutions for an adoption of agile methods in automotive software development
<p><u>Validated challenges:</u></p> <ul style="list-style-type: none"> - Strict design requirements engineering processes that constrain the flexibility of Scrum - Assurance of regulatory compliance by an incorporation of complex safety requirements into Scrum and Continuous Integration development processes - Long lead times between change commits and acceptance, due to extensive software testing and verification processes when working with Continuous Integration - Organizational cultural deficiencies in regard of a permission for necessary decision authority of the development team 	<p><u>Additional challenges:</u></p> <ul style="list-style-type: none"> - Uncertainty, during Scrum sprints in automotive software development, which constraints plannability and ability to make accurate forecasts, as well as confident stakeholder reporting - Delays, caused by the difficulty of synchronizing hardware and software development
<p><u>Validated solutions:</u></p> <ul style="list-style-type: none"> - Usage of hybrid processes to run Scrum within the V-Model - Creation of a solid DoD to ensure the fulfillment of requirements within a Scrum sprint - Transparency regarding the completed content at the end of a Scrum sprint - Implementation of a master repository and quality gates, as well as automated tests, to ensure a stable level of quality when working with Continuous Integration - Establishment of shared team goals, to increase the motivation of the development teams 	<p><u>Additional solutions:</u></p> <ul style="list-style-type: none"> - Application of Scrum at the bottom part of the V-Model, without a fixed sequential order of requirements engineering and testing procedures - Replacement of strict sequential processes, by the DoD as flexible quality assurance mechanism - Establishment of an organizational culture that teaches the employees to view agile as a mindset that has to be adopted

Table 4. Validated and additional Challenges and Solutions for an Adoption of Agile Methods in Automotive Software Development based on Expert Interviews

7. SUMMARY AND CONCLUSION

The objective of this research is to provide an overview of current challenges and solution approaches for the adoption of agile methods in automotive software development. To answer the aforementioned research question “*How can automotive manufacturers, in light of a demand for more flexible vehicle software development processes, adopt Agile Methods?*”, a literature review and structured interviews with employees of a German automotive manufacturer have been conducted.

With the literature review, this paper provided an overview of current challenges and solutions for the adoption of agile methods in automotive software development, that have been discussed by researchers in the past three years. Out of numerous agile methods, Scrum and Continuous Integration have been found to be applied the most frequently in automotive software development (Kulger Maag Cie, 2015). There are various safety related constraints, amongst other things, safety standards like the ISO26262 and the implicitly suggested automotive SPICE quality assessment framework, which demand various requirements and tests (Komiyama et al., 2019, Van der Valk et al., 2018). As the adoption of agile methods also involves the people, who may have to operate in a different way than before, also organizational cultural barriers have been discovered. In regard of this, studies indicate that managers are hesitant to provide the operative levels with sufficient decision authority, even though this is needed for agile software development (Marner et al., 2019). Nevertheless, there are also various solution approaches to adapt automotive software development processes and incrementally change the corporate culture. In general the literature for safety challenges of Scrum suggests to create hybrid structures, which are a mix of traditional waterfall processes and agile methods, to make use of the benefits of Scrum, but also ensure that safety requirements are met (Komiyama et al., 2019). Furthermore, it was pointed out that an incremental adoption of agile methods, especially Scrum, is advised to increase the organizational tolerance for the new way of working stepwise (Hagihghatka, 2017; Diebold and Theobald, 2018; Komiyama, 2019). As Continuous Integration in automotive software development can only be used under the condition of a guarantee for a certain level of software quality, literature suggests to implement quality gates that ensure that software changes are checked and approved, before they are added to the master repository, which is the stable basis for the software. Next to that, collaboration was also stated to be an important aspect for the efficient usage of Continuous Integration. In that regard, it was advised to increase the level of transparency, in form of shared information, between organizations that operate in the same value chain, as well as the introduction of agile supplier contracts (Van der Valk et al., 2018; Pellicione et al., 2017).

An explorative research consisting of five interviews with industry experts was conducted to complement and validate the literature-based answer of the research question. In line with this objective there were five structured interviews with employees of a German automotive manufacturer, in which the experts’ general experiences and perceptions, as well as the solution approaches for an adoption of agile methods in automotive software development were explored. The analysis of the expert interviews showed that the experts have encountered different challenges that constrain the usage of Scrum in automotive software development, and for Continuous Integration there were disagreements whether there are challenges at all. As the literature on agile methods in automotive software development mostly points out safety concerns, safety was a prioritized topic during the interviews. However, contrary to other topics, all the experts agreed that safety and quality requirements are important,

but that there is not a mandatory development model, at least for the solely software related development processes, to assure that the requirements are fulfilled within the Scrum development process. According to the experts, the most important factor is that the requirements are met in the end but there does not have to be a strict order in which quality mechanisms, such as software unit testing, have to be conducted. Furthermore, the experts also agree and validate the literature findings, that safety during Continuous Integration can be ensured by strict test automation and quality gates that prevent developers to add software codes, which do not meet the requirements, to the master repository. Additionally, contrary to the literature, in which organizational cultural factors were mentioned only partially, multiple experts pointed out that trust, motivation and agile as a mindset, are essential aspects for an adoption of agile automotive software development methods. One expert even stated that these organizational cultural factors are currently often times missing in practice and that agile methods, like Scrum, provide another opportunity for managers to control the software development team.

The provided research indicates there are many aspects that have to be considered when using agile methods in automotive software development, and, contrary to the methodology on traditional software development processes, such as the V-Model, there is not a shared guideline on how to apply agile methods, such as Scrum and Continuous Integration.

Based on the aforementioned aspects, to answer the research question “*How can automotive manufacturers, in light of a demand for more flexible vehicle software development processes, adopt Agile Methods?*”, we could say that agile methods cannot be used by the book, but have to be adapted to the automotive software development environment, especially to functional safety standards such as the ISO26262. Even though there is not a strict development model, it is important that at the end of each iteration, there is a product increment, that fulfills specific requirements and is potentially shippable. The literature on agile automotive software development provides different solutions to ensure quality, regulatory compliance, as well as recommendations to improve collaboration internally and externally. Additionally, an important aspect, that seems to be understated in scientific literature, but was emphasized by industry experts, is to establish an organizational culture that understands agile not only as a process but as a mindset. This mindset is anchored within the core values of the agile software development idea and is essential to create an added value for automotive software development, by enabling more flexibility and shorter software release cycles.

8. LIMITATIONS AND FUTURE RESEARCH SUGGESTIONS

8.1 Limitations

This study comes with several limitations that are important to note. To begin with, only five expert interviews, which is a relatively small sample size, were conducted and therefore, the qualitative data can only provide indications of general tendencies. Furthermore, the interviewees did all work for the same company, which means that additional challenges and solution approaches for the adoption of agile methods in the automotive industry, encountered by other companies, might be missing. However, conducting more interviews with experts from different companies and different organizational levels, were out of scope and time for this paper. To increase the validity of this data collection method, the amount of expert interview partners should be increased. Next to the issue of the sample size, there are also limitations regarding the questions that were asked during the interviews. Since there were time constraints, the focus

of the interview question was predominantly on safety-related topics, which means that in depth question about organizational structural, and collaboration challenges and solution approaches for an adoption of agile methods in automotive software development, were missing. Based on this issue, it was also not possible to further investigate the challenge of supplier contracting that was discovered during the literature review on Continuous Integration. As suppliers play an important role for Continuous Deployment, which takes place after the Continuous Integration process and is also responsible for the frequency in which new product functionalities and updates can be delivered to the customers, this is a substantial barrier to which this paper does not provide a solution approach that incorporates inputs from industry experts.

As shortly elaborated within the theoretical framework, there are also other agile methods besides Scrum and Continuous Integration. However, even after a conduction of intensive desk research, it was not possible to find sufficient qualitative scientific literature, to explore these methods as part of this research project.

8.2 Future Research Suggestions

There are also several suggestions for interesting research in the future. One suggestion for future research can be derived from the results of two expert interviews, which indicated that there is not a shared consensus on challenging factors regarding the usage of Continuous Integration. While a software developer, who deals with Continuous Integration on a daily base, says the following about Continuous Integration in automotive software development:

- *“This Continuous Integration is mostly horrible because of the ISO aspects because of all the legal requirements it has inflated to a point where it is not manageable. The ISO really creates a problem for Continuous Integration, and I have been able to talk to many people from different departments and also other companies.”* – E5

A manager perceives Continuous Integration the opposite way:

- *“For me Continuous Integration is having a system ready that is performing your built-in tests as well and being able to monitor at any given time. In that regard I would say that it is as easy as in every other industry.”* – E2

Based on these two opposing statements, it could be interesting to investigate if there are different perceptions on challenges and the usefulness of solution approaches for an adoption agile method in automotive software development, depending on the hierarchical level of different respondents. However, the issue of functional safety-standards, in this case the ISO26262, that decrease the speed of Continuous Integration processes, has also been discovered within the literature review. Nevertheless, neither the literature review, nor the expert interviews, provided a solution approach on how to remain efficient, while assuring regulatory compliance at the same time. This could be the basis for a research on further ways to deal with this efficiency challenge, but also whether all of the safety requirements of the ISO26262 are actually necessary.

Finally, another research suggestion would be to investigate how other agile methods besides Scrum and Continuous Integration can be adopted, as there is currently only limited scientific literature available.

9. RESEARCH CONTRIBUTION

9.1 Contribution to Theory

As agile methods are not a new concept, there are many scholars that have discussed general challenges and implementation approaches for agile software development methods in different domains. However, in the context of automotive, which is a regulated domain, there are predominantly contributions that address specific challenges and solutions for certain agile methods. Nevertheless, there is not commonly shared approach for and adoption of agile methods in the automotive domain, nor is there a contribution that provides an overview over current challenges and solution approaches for an adoption of agile methods, such as Scrum and Continuous Integration. This research project, on the one hand, combines the challenges and solution approaches that have been discussed by different scholars, and on the other hand, validates and complements these contributions by adding practical experiences of industry experts. Therefore, this research can also be viewed as a first step to create a shared understanding on how agile methods can be adopted for the development of automotive software functionalities.

9.2 Contribution to Practice

In practice the added value of this research derives from the opportunity for automotive manufacturers to reevaluate their current approach of dealing with the agile transformation of software development processes. As elaborated in the problem statement, the competitive landscape within the automotive industry is rapidly changing and software release cycles have to become shorter to remain competitive in the future. As agile methods have been found to reduce the release cycle time, this research project provides automotive manufacturers with insights of literature-indicated challenges and solutions for the adoption agile software development methods. This means there might be unheard solution approaches that could add some value to the current agile automotive software development approach. Furthermore, the expert interviews included employees from different organizational levels. As a matter of that, this research project provides an opportunity for automotive manufacturers to assess current challenges from different, important, perspectives, as it is sometimes difficult to obtain an overview of challenges that different organizational levels are dealing with, during the daily business.

10. ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisors Dr. Matthias de Visser, Liza Kayser, and Dr. Fons Wijnhoven for their tremendous supervision and support that they offered me during the process of this research project. Even though this research project was conducted in the middle of a global crisis, which made it impossible to meet each other in person, you were always available to discuss my progress and your feedback was crucial for me to improve important aspects of this thesis. Last, but not least, I would like to thank my parents for constantly supporting me during the entire International Business Administration study, without whom nothing would have been possible. Lastly, I would also like to express my appreciation to you, the reader, who took the time to read my work.

11. REFERENCES

- Agren, S. M., Knauss, E., Heldal, R., Pelliccione, P., Malmqvist, G., & Boden, J. (2019). The impact of requirements on systems development speed: a multiple-case study in automotive. *Requirements Engineering*, 24(3), 315-340. doi:10.1007/s00766-019-00319-8
- Bach, J., Langner, J., Otten, S., Holzäpfel, M., & Sax, E. (2017). Data-Driven Development, A Complementing Approach for Automotive Systems Engineering.
- Banijamali, A., Dawadi, R., Ahmad, M. O., Similä, J., Oivo, M., & Liukkunen, K. (2017). Empirical Investigation of Scrum in Global Software Development.
- Beck, K. (1999). Embracing change with extreme programming. *Computer*, 32(10), 70-77. doi:10.1109/2.796139
- Diebold, P., & Theobald, S. (2018). How is agile development currently being used in regulated embedded domains? *Journal of Software: Evolution and Process*, 30, e1935. doi:10.1002/smr.1935
- Diebold, P., Zehler, T., & Richter, D. (2017). How do agile practices support automotive SPICE compliance? Paper presented at the ACM International Conference Proceeding Series.
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10), 833-859. doi:10.1016/j.infsof.2008.01.006
- Fahl, J., Hirschter, T., Kamp, J., Endl, M., & Albers, A. (2019). Functional concepts in the model of PGE-product generation engineering by the example of automotive product development. Paper presented at the ISSE 2019 - 5th IEEE International Symposium on Systems Engineering, Proceedings.
- Gaiamo, F., Andrade, H., & Berger, C. (2019). The Automotive Take on Continuous Experimentation: A Multiple Case Study. Paper presented at the Proceedings - 45th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2019.
- Goodman, D., & Elbaz, M. (2008, 4-8 Aug. 2008). "It's Not the Pants, it's the People in the Pants" Learnings from the Gap Agile Transformation What Worked, How We Did it, and What Still Puzzles Us. Paper presented at the Agile 2008 Conference.
- Haghighatkah, A., Oivo, M., Banijamali, A., & Kuvaja, P. (2017). Improving the State of Automotive Software Engineering. *IEEE Software*, 34(5), 82-86. doi:10.1109/MS.2017.3571571
- Hantke, D. (2015). An Approach for Combining SPICE and SCRUM in Software Development Projects, Cham.
- Heerwagen, M. (2018). The Changing Face of Development Agile Methods Gain Ground. *MTZ worldwide*, 79(4), 8-13. doi:10.1007/s38313-018-0027-3
- Hohl, P., Stupperich, M., Munch, J., & Schneider, K. (2018). An Assessment Model to Foster the Adoption of Agile Software Product Lines in the Automotive Domain. Paper presented at the 2018 IEEE International Conference on Engineering, Technology and Innovation, ICE/ITMC 2018 - Proceedings.
- Kassner, L., Groger, C., Königsberger, J., Hoos, E., Kiefer, C., Weber, C., . . . Mitschang, B. (2017). The Stuttgart IT Architecture for Manufacturing An Architecture for the Data-Driven Factory. In S. Hammoudi, L. A. Maciaszek, M. M. Missikoff, O. Camp, & J. Cordeiro (Eds.), *Enterprise Information Systems*, Iccis 2016 (Vol. 291, pp. 53-80). Berlin: Springer-Verlag Berlin.
- Katumba, B., & Knauss, E. (2014). Agile Development in Automotive Software Development: Challenges and Opportunities.
- Kim, J., Hwangbo, H., & Kim, S. (2018). An empirical study on real-time data analytics for connected cars: Sensor-based applications for smart cars. *International Journal of Distributed Sensor Networks*, 14(1), 13. doi:10.1177/1550147718755290
- Klunder, J., Hohl, P., & Schneider, K. (2018). Becoming Agile While Preserving Software Product Lines An Agile Transformation Model For Large Companies. New York: Assoc Computing Machinery.
- Komiyama, T., Konno, S., Watanabe, T., Matsui, S., Kase, M., & Igarashi, I. (2019) Improvement of Agile Software Development Process Based on Automotive SPICE: A Case Study. In: Vol. 1060. *Communications in Computer and Information Science* (pp. 518-531).
- Kuhrmann, M., Diebold, P., Munch, J., Tell, P., Trektere, K., McCaffery, F., . . . Prause, C. R. (2019). Hybrid Software Development Approaches in Practice: A European Perspective. *IEEE Software*, 36(4), 20-31. doi:10.1109/MS.2018.110161245
- Kugler Maag CIE (2015). Agile in Automotive - State of Practice 2015.
- Liu, B. H., Zhang, H., & Zhu, S. C. (2016). An Incremental V-Model Process for Automotive Development. In A. Potanin, G. C. Murphy, S. Reeves, & J. Dietrich (Eds.), 2016 23rd Asia-Pacific Software Engineering Conference (pp. 225-232). New York: Ieee.
- Marner, K., Theobald, S., & Wagner, S. (2019). Real-life challenges in automotive release planning. Paper presented at the Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019.
- Obergfell, P., Kugele, S., Segler, C., Knoll, A., Sax, E., & Ieee. (2019). Continuous Software Engineering of Innovative Automotive Functions: An Industrial Perspective. New York: Ieee.
- Pelliccione, P., Knauss, E., Heldal, R., Magnus Ågren, S., Mallozzi, P., Alminger, A., & Borgentun, D. (2017). Automotive Architecture Framework: The experience of Volvo Cars. *Journal of Systems Architecture*, 77, 83-100. doi:10.1016/j.sysarc.2017.02.005
- Pfeffer, R., Basedow, G., Thiesen, N., Spadinger, M., Albers, A., & Sax, E. (2019). Automated Driving - Challenges for the Automotive Industry in Product Development with Focus on Process Models and Organizational Structure.
- Ribeiro, F., Ferreira, A. L., Tereso, A., & Perrotta, D. (2018) Development of a grooming process for an agile software team in the automotive domain. In: Vol. 745. *Advances in Intelligent Systems and Computing* (pp. 887-896).
- Sandu, I. A., & Salceanu, A. (2019). System Testing in Agile SW Development of the Electronic Components Based on Software from the Automotive Industry. Paper presented at the 2019 11th International Symposium on Advanced Topics in Electrical Engineering, ATEE 2019.
- Sax, E., Reussner, R., Guissouma, H., & Klare, H. (2017). A Survey on the State and Future of Automotive Software Release and Configuration Management. Retrieved from <http://dx.doi.org/10.5445/IR/1000075673>
- Schlichthärle, S., Becker, K., & Sperber, S. (2020). A Domain-Specific Language Based Architecture Modeling

- Approach for Safety Critical Automotive Software Systems.
- Soares, H. F., Alves, N. S. R., Mendes, T. S., Mendonça, M., & Spínola, R. O. (2015, 13-15 April 2015). Investigating the Link between User Stories and Documentation Debt on Software Projects. Paper presented at the 2015 12th International Conference on Information Technology - New Generations.
- Ståhl, D., & Bosch, J. (2014). Modeling continuous integration practice differences in industry software development. *Journal of Systems and Software*, 87(1), 48-59. doi:10.1016/j.jss.2013.08.032
- Steghöver, J. P., Knauss, E., Horkoff, J., & Wohlrab, R. (2019) Challenges of Scaled Agile for Safety-Critical Systems. In: Vol. 11915 LNCS. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 350-366).
- Takahira, R. Y., Laraia, L. R., Dias, F. A., Yu, A. S., Nascimento, P. T. S., & Camargo, A. S., Jr. (2014). Scrum and Embedded Software development for the automotive industry. Paper presented at the PICMET 2014 - Portland International Center for Management of Engineering and Technology, Proceedings: Infrastructure and Service Integration.
- Van Der Valk, R., Pelliccione, P., Lago, P., Heldal, R., Knauss, E., & Juul, J. (2018). Transparency and contracts: Continuous integration and delivery in the automotive ecosystem. Paper presented at the Proceedings - International Conference on Software Engineering.
- Vost, S., & Wagner, S. (2017, 20-28 May 2017). Keeping Continuous Deliveries Safe. Paper presented at the 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C).

12. APPENDIX

12.1 Interview Questions for the Experts

Requirements for the interview:

- Interviewee has participated in an agile automotive software development project
- Interviewee has dealt with Scrum

1: What is your role in an agile team?

2: What was your biggest challenge when dealing with Scrum in an automotive software development project and how did you approach it?

3: Have you worked with Continuous Integration, and if yes, what are your experiences with Continuous Integration in automotive software development?

4: How can Scrum teams handle automotive SPICE requirements in regard of specification and testing? (followed by a short explanation of what is meant with automotive SPICE requirements, and if unknwon, the question if there are other known quality assurance mechanisms)

5: How can a certain level of quality be ensured when applying Continuous Integration?

6: Literature indicates that Scrum cannot be “used by the book” but must be adapted to the automotive software development context. Which elements of Scrum are applicable, and which traditional (waterfall) processes are indispensable for automotive software development?