

# Forces that Prevent Agile Adoption in the Automotive Domain

Philipp Hohl<sup>1(✉)</sup>, Jürgen Münch<sup>2,3</sup>, Kurt Schneider<sup>4</sup>, and Michael Stupperich<sup>1</sup>

<sup>1</sup> Daimler AG, Research and Development, Ulm, Germany  
{philipp.hohl,michael.stupperich}@daimler.com

<sup>2</sup> Reutlingen University, Reutlingen, Germany  
juergen.muench@reutlingen-university.de

<sup>3</sup> University of Helsinki, Helsinki, Finland

<sup>4</sup> Leibniz Universität Hannover, Hannover, Germany  
kurt.schneider@inf.uni-hannover.de

## Abstract.

**Context:** The current transformation of automotive development towards innovation, permanent learning and adapting to changes are directing various foci on the integration of agile methods. Although, there have been efforts to apply agile methods in the automotive domain for many years, a wide-spread adoption has not yet taken place.

**Goal:** This study aims to gain a better understanding of the forces that prevent the adoption of agile methods.

**Method:** Survey based on 16 semi-structured interviews from the automotive domain. The results are analyzed by means of thematic coding.

**Results:** Forces that prevent agile adoption are mainly of organizational, technical and social nature and address inertia, anxiety and context factors. Key challenges in agile adoption are related to transforming organizational structures and culture, achieving faster software release cycles without loss of quality, the importance of software reuse in combination with agile practices, appropriate quality assurance measures, and the collaboration with suppliers and other disciplines such as mechanics.

**Conclusion:** Significant challenges are imposed by specific characteristics of the automotive domain such as high quality requirements and many interfaces to surrounding rigid and inflexible processes. Several means are identified that promise to overcome these challenges.

**Keywords:** Software development · Agile methods · Automotive

## 1 Introduction

The automotive industry is confronted with high-frequent changes due to innovations and new technology. Today, it is a competitive advantage for car manufacturers to develop and distribute high-quality software at a high pace. A promising solution to keep pace with this progress are agile software development methods.

High quantities with high cost pressure, test and validation under real-time conditions and a high amount of software variants are important characteristics for automotive software development. Safety-critical applications on the one hand and cost pressure on the used hardware on the other hand are encountered in the automotive domain. Furthermore, strict processes for car development have to be considered. Automotive functions must be verified by long term field tests and endurance tests that are enforced by law.

This survey investigates the specifics of agile adoption in the automotive domain. It presents an interview-based qualitative survey that aims to understand today's state of the practice of perceived forces on agile adoption. This work focuses on the agile software development for electronic control units (embedded software) in the automotive domain.

## 2 Related Work

Since a decade and more, agile software development methods show promising benefits in domains such as mobile or web development. In the beginning, it was not yet clear if agile software development would be applicable to the automotive field. In 2004, Manhart and Schneider [1] “tried to break the ice” for agile embedded development. They summarized existing experiences with agile methods, but emphasized that more knowledge of agile practices is needed. A survey by Kugler Maag CIE [2] in 2015, revealed a non-uniform adoption of agile development in the automotive domain with various and selective adopted agile methods and practices.

A leader in agile adoption is the Original Equipment Manufacturer (OEM) Volvo. Eliasson et al. [3] performed several case studies at Volvo to identify the limitation of agile development in the automotive domain. These studies focus on impacts with respect to software architecture and pointed out cooperation problems with subcontractors. In 2014, Eliasson et al. [4] identified the necessity to reveal possible showstoppers in the earlier phases of projects by means of faster feedback. In addition, they focus on requirements engineering combined with agile practices. Stelzmann et al. [5] analyzed success factors which help projects to become agile. Katumba et al. [6] conducted a case study to identify challenges in software development process related to frequent task switching, individualism, lack of complete knowledge and communication. With this in mind, our study aims at identifying hindering forces on the agile adoption and potential solution approaches.

## 3 Study Approach

### 3.1 Research Questions

RQ 1: What are the perceived forces that prevent agile adoption in automotive?

RQ 1.1: What are the habits and inertia that prevent agile adoption?

RQ 1.2: What are the anxiety factors that prevent agile adoption?

RQ 1.3: Which context factors prevent agile adoption?

RQ 2: What are the perceived means to adopt agile in automotive?

### 3.2 Research Design

The study is based on a qualitative survey. It is designed as an exploratory semi-structured interview. The method provides insights into the examined topic and gives essential information to understand the phenomenon in its real context [7,8]. For a semi-structured interview an interview guide was implemented [9]. The interview guide was structured along a funnel model [8]. Each section begins with open, exploratory ground mapping questions [10]. These questions reveal all topics of interests [11]. In addition, dimension mapping questions are used to focus on interesting topics [11]. The interview guide was tested in a pilot interview and adjusted to the problems which have arisen.

### 3.3 Data Collection and Analysis

**Research Sites and Participants.** The interview participants were selected from employees of an OEM and an automotive consultant. The interviewee selection was based on two criteria: First, the interviewee should have a work experience of several years. The length of employment varied from 3 to 20 years, with an average working experience of 16 years. Second, the interviewee should already use agile practices for software development. To get a different point of view on the examined topic, the following participants were selected: Two managers, five process owners, two system architects, six software developers and one automotive consultant for agile development processes. The interviews were conducted by the primary researcher at the interviewees departments from May to June 2016.

**Interviews.** There were 14 face-to-face interviews as well as a group interview with two participants. Every interview took around one hour. The interview questions were initially defined in English and translated to the native language of the interviewees. In consent with the interviewee, the interview was recorded and transcribed verbatim for detailed analysis. All transcribed interviews notes were managed using the reference management program Citavi.

**Analysis.** According to the classification of Stol et al. [12], the coding concepts of Straussian Grounded Theory were used. We used the three coding phases of Straussian Grounded Theory: open coding, axial coding, and selective coding [12, 13]. The interpretive process of open coding generates categories and concepts by breaking down the data analytically. The concepts were grouped together and related to their subcategories in the axial coding. In the selective coding the central categories were defined.

**Validity Considerations.** Validity was threatened by the possibility of misunderstandings between interviewees and the researcher. To minimize this risk, the study goal was explained to the participants prior to the interview. Steps taken to improve the reliability of the interview guide included a review and a pilot test. To reduce researcher bias, the interviews were recorded and transcribed.

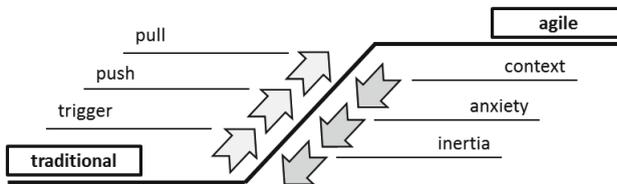
## 4 Results

### 4.1 Forces on Agile Adoption

We define six categories of forces on agile adoption (cf. Fig. 1). The categorization aims to better understand the different aspects of the transition process from traditional to agile development practices. We distinguish between “trigger”, “push” and “pull” as forces that lead to agile adoption. In contrast, we define “inertia”, “anxiety”, and “context” as forces that prevent the agile adoption. The classification is inspired by the Customer Forces Diagram by Maurya<sup>1</sup> that itself is inspired by the Forces Diagram by Moesta and Spiek from the Jobs-to-be-done framework<sup>2</sup>.

A trigger force initiates a change and pushes an individual or an organization towards agile adoption. A push force pushes an individual or an organization towards adopting agile practices based on issues or demands. A pull force come into effect when individuals or organizations are pulled towards agile adoption based on the attractiveness of a future situation.

Inertia forces, such as habits, keep people from trying out something new and hence prevents agile adoption. The anxiety forces are representing fears which could prevent the adoption of agile. Often, uncertainties surrounding new situations cause anxiety. The context forces result from constraints and obstacles in the environment (such as organizational structures or process barriers) and prevent agile adoption. In the following, the preventing forces found in the study are subsequently described and explained using the information from the interview transcripts and the interpretations from coding and analysis.



**Fig. 1.** Agile adoption forces diagram

<sup>1</sup> <https://leanstack.com/science-of-how-customers-buy/>.

<sup>2</sup> <http://jobstobedone.org>.

## 4.2 Perceived Forces that Prevent Agile Adoption

**Inertia.** Most of the interviewees mention that a major problem is the missing understanding of the applicability of agile methods within their context. Some interviewees emphasize that it is not clear for the management how to manage agile development and how to integrate it into their departments. Managers point out that a change of the mind-set is needed to adopt agile, but it is unclear how to achieve this. All interviewees mention that applying agile methods might require more communication effort. 50% of the interviewees agree that communication effort is manageable in local, small teams but it is difficult to coordinate if the development is distributed.

**Anxiety.** The developers believe that it is necessary in agile development to give more responsibility to software developers; they mention that the management does not want to give up responsibilities. The managers emphasize that it is unclear how to provide correct estimations on development efforts when applying agile practices. They mention that it is difficult to prioritize features without correct estimations. In addition, the managers mention that they do not want to displease software developers by changing roles and responsibilities. In fact, they fear that annoyed developers might leave the department. The software developers emphasize their biggest fear that customer-relevant defects remain in the delivered software.

**Context.** Except of two, all interviewees mention that with the current structure of the company too many responsible persons are involved in negotiations about feature implementation. The interviewees state that this slows down the software development and prevents agile adoption. Most of the interviewees consider the high amount of process-dependencies for software development as an impediment for the transition towards agile development. One manager mentions the demand for more employees to maintain and manage the intersection between the agile department and the traditional organization. The processes on the higher system levels are seen as important. At the same time, however, they are considered to prevent and restrict agile development. The software developers emphasize the need to synchronize with fixed freeze dates and hardware-development what is seen as slowing down the process. In addition, one interviewee attributes the longer development time to the increased communication effort during the implementation. The interaction with the purchasing department and suppliers is highlighted by most interviewees as a challenging task. The communication with a supplier is identified to be a problem. Challenges with respect to communication are as well present in the context of globally distributed software development projects.

All interviewees mention the high effort to fulfill compliance and validation. Technical risks and challenges are mentioned by seven interviewees. It is important that the software is validated and of high quality. The developers mention that test and validation departments cannot increase speed due to the necessity

of integration and validation in a real car. All interviewees refer to limited capacity in manpower and test systems when it comes to validation. Therefore, it is necessary to reuse software parts in order to reduce certification efforts. Other restrictions which are seen as important for adopting agile are long term field tests and endurance runs that are enforced by law, e.g. summer and winter tests. These tests must be kept at reasonable costs.

### 4.3 Perceived Means to Adopt Agile

**How to Overcome the Inertia?** Most of the participants mention that they already use incremental builds to shorten the release cycles. In addition, the interviewees stress that they have implemented approaches to build prototypes independent from the main development. Therefore they introduced auxiliary processes to provide an environment for faster internal releases.

The managers mention that more than 80% of the software development for selected electronic control units is already transferred to in-house development. One developer emphasizes that in-house development is appropriate if the specification effort for a feature functionality exceeds the development effort. Another interviewee describes a situation in which in-house implementation is not possible. He stresses that the collaboration with the supplier should be a closer collaboration with more coordination and communication.

**How to Overcome the Anxiety?** The developers mention an increasing software quality at an early stage of the development is allaying their fear of late defects in the software. An increase in learning speed is considered as a mean to increase the odds for delivering high quality software. Other interviewees expect that prototyping might help to create a safe to fail environment that allows a fast feedback about a new function which is under development. In addition, the risks of late defects can be reduced.

Several interviewees mention that software developers should be granted more responsibility. One manager refers to the one-room principle where different employees from different engineering domains like software, electronics and mechanics are sitting together in one office. He highlights the benefits from interdepartmental cooperation. A mind-set change and redistribution of responsibility is therefore necessary to keep a cooperative atmosphere.

**How to Overcome Obstacles Imposed by the Context?** All interviewees identified organizational structures in large organizations as a main reason of preventing the adoption of agile practices. They stress that it is almost impossible to change the organization. Three participants mention that although top management is fostering the change towards agile development, this is not the case on all management levels. They recommend reorganization with lower hierarchies.

One interviewee describes how his department embedded agile software development in small agile environments in the organization. He emphasizes that this

approach is manageable on a small scale but needs more employees to maintain the intersection with the traditional organization. The interviewee mentions the need for new processes in order to manage the interaction between the traditional organization and the agile environment. Two participants mention that they recruited a consultant to adapt agile practices appropriately to the context of their department.

Several interviewees state that the benefits achieved so far by the use of product line management should not be neglected. Therefore, one participant emphasizes that the level of software reuse should be maintained and extended by the use of simulations that replace unavailable parts during development. Table 1 summarizes the challenges with associated possible solution approaches for the forces that prevent agile adoption.

**Table 1.** Challenges and Solution approaches

Forces	Challenge	Solution approach
Inertia	Missing understanding of the applicability of agile methods in a specific context	Organize context-specific agile training and coaching
	A change of the mind-set is needed to adopt agile practices	Collaborate with (external) experts
	More communication effort required	Word-of-mouth recommendations and feedback-culture
	Current development process is seen as satisfying	Explain benefits of agile adoption
	Limited acceptance for organizational restructuring	Slow and stepwise integration of agile methods
Anxiety	Management does not want to give up responsibilities	Redefine management role; Software developers should be granted more responsibility
	Unclear how to provide correct estimations on development efforts when applying agile practices	Shorten estimation interval
	Fear that customer-relevant defects remain in the delivered software	Provide a safe to fail environment and prototype development
Context	Rigid and inflexible surrounding processes	Define interface between agile and traditional processes
	Hierarchy in the organization	Transform hierarchies into networks
	Frequent synchronization with suppliers and QA	More in-house software development
	Globally distributed software development	Use of simulations that replace unavailable parts
	Keep benefits from software reuse	Reorganize product line development

## 5 Discussion and Conclusion

The study presented a state of the practice analysis on agile adoption in the automotive domain. In 2015, Kugler Maag CIE [2] revealed a selective adoption of agile methods in the automotive domain. In addition, our study identified the forces on the agile adoption. Furthermore, our study associates the findings by Katumba [6] and Eliasson [3,4]. Key challenges in agile adoption are related to transforming organizational structures and culture, achieving faster software release cycles without loss of quality, the importance of software reuse in combination with agile practices, appropriate quality assurance measures and the collaboration with suppliers.

The survey reveals many avenues for further research. Potential directions could be to integrate agile practices into existing product lines in the automotive domain and to identify means for addressing the restrictions of a rather strict surrounding process.

## References

1. Manhart, P., Schneider, K.: Breaking the ice for agile development of embedded software: an industry experience report. In: Proceedings of the 26th International Conference on Software Engineering, pp. 378–386, 23–28 May 2004
2. Weber, S.: Agile in Automotive – State of Practice (2015). [www.kuglermaag.com/agile2015](http://www.kuglermaag.com/agile2015). Accessed 1 Dec. 2015
3. Eliasson, U., Heldal, R., Pelliccione, P., Lantz, J.: Architecting in the automotive domain: descriptive vs prescriptive architecture. In: Bass, L., Lago, P., Kruchten, P. (eds.) 12th Working IEEE/IFIP Conference on Software Architecture, WICSA, pp. 115–118. IEEE, Piscataway (2015)
4. Eliasson, U., Heldal, R., Lantz, J., Berger, C.: Agile model-driven engineering in mechatronic systems - an industrial case study. In: Dingel, J., Schulte, W., Ramos, I., Abrahão, S., Insfran, E. (eds.) MODELS 2014. LNCS, vol. 8767, pp. 433–449. Springer, Heidelberg (2014). doi:10.1007/978-3-319-11653-2\_27
5. Stelzmann, E., Kreiner, C., Spork, G., Messnarz, R., Koenig, F.: Agility meets systems engineering: a catalogue of success factors from industry practice. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 245–256. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15666-3\_22
6. Katumba, B., Knauss, E.: Agile development in automotive software development: challenges and opportunities. In: Jedlitschka, A., Kuvaja, P., Kuhrmann, M., Männistö, T., Münch, J., Raatikainen, M. (eds.) PROFES 2014. LNCS, vol. 8892, pp. 33–47. Springer, Heidelberg (2014). doi:10.1007/978-3-319-13835-0\_3
7. Dresch, A., Lacerda, D.P., Antunes, J.A.V.: Design Science Research. Springer International Publishing, Cham (2015)
8. Runeson, P., Hst, M.: Guidelines for conducting and reporting case study research in software engineering. *Empirical Softw. Eng.* **14**(2), 131–164 (2009)
9. Bryman, A.: *Social Research Methods*, 2nd edn. Univ. Press, Oxford (2004)
10. Easterbrook, S., Singer, J., Storey, M.-A., Damian, D.: Selecting empirical methods for software engineering research. In: Shull, F., Singer, J., Sjberg, D.I.K. (eds.) *Guide to Advanced Empirical Software Engineering*, pp. 285–311. Springer-Verlag, London Limited, London (2008)

11. Ritchie, J. (ed.): *Qualitative Research Practice: A Guide for Social Science Students and Researchers*. Sage, repr ed., Los Angeles (2011)
12. Stol, K.-J., Ralph, P., Fitzgerald, B.: Grounded theory in software engineering research. In: Dillon, L., Visser, W., Williams, L. (eds.) *Proceedings of the 38th International Conference on Software Engineering*, pp. 120–131 (2016)
13. Corbin, J., Strauss, A.: Grounded theory research: procedures, canons and evaluative criteria. *Qual. Sociol.* **13**, 3–21 (1990)