Real-Life Challenges on Agile Software Product Lines in Automotive

Philipp Hohl[1]([✉]), Jürgen Münch[2], Kurt Schneider[3], and Michael Stupperich[1]

[1] Daimler AG, Research and Development, Ulm, Germany
{philipp.hohl,michael.stupperich}@daimler.com
[2] Reutlingen University, Reutlingen, Germany
juergen.muench@reutlingen-university.de
[3] Leibniz Universität Hannover, Hanover, Germany
kurt.schneider@inf.uni-hannover.de

Abstract. Context: The current situation and future scenarios of the automotive domain require a new strategy to develop high quality software in a fast pace. In the automotive domain, it is assumed that a combination of agile development practices and software product lines is beneficial, in order to be capable to handle high frequency of improvements. This assumption is based on the understanding that agile methods introduce more flexibility in short development intervals. Software product lines help to manage the high amount of variants and to improve quality by reuse of software for long term development.

Goal: This study derives a better understanding of the expected benefits for a combination. Furthermore, it identifies the automotive specific challenges that prevent the adoption of agile methods within the software product line.

Method: Survey based on 16 semi-structured interviews from the automotive domain, an internal workshop with 40 participants and a discussion round on ESE congress 2016. The results are analyzed by means of thematic coding.

Results: Two main expected benefits of merging agile practices and product line development are pushing the change in software development for future proof agile automotive organizations. Challenges that prevent agile adoption within software product lines are mainly of organizational, technical and social nature. Key challenges are related to transforming organizational structures and culture, achieving faster software release cycles without loss of quality, appropriate quality assurance measures for software variants, 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.

Keywords: Automotive · Agile software development · Software product line

1   Introduction

The automotive domain is recently in a disruptive change. High-frequent changes due to innovations and new technology confront the automotive domain. Several possible future scenarios are becoming apparent. One scenario comprises the changeover from the internal combustion engine to electric and hydrogen fuel cell cars [1]. New competitors entering the market, pushing the traditional car manufactures to react on the changing market situation. The situation is further exacerbated by political agenda, e.g., german politics requests for 1 million electric cars within 2020[1]. The plan is now abandoned because at present it is considered unrealistic[2]. Nevertheless, a lot of suppliers are very active in this development area, encourage the traditional car manufactures to make more effort to achieve at least the same development pace. It is seen as a necessity to be competitive in future. Another scenario comprises self-driving, community owned cars. This will be a further disruptive change in the domain, because car-sharing has a direct impact on the ownership of private cars. The scenario implies that anyone can order a car wherever and whenever it is needed. Several challenges arise with this form of car-sharing, like challenges in high-precise navigation and autonomous driving. The solution of these challenges will all be addressed in software. The described scenarios are likely to exist in parallel. This does not solely affect the entire car transportation system, but the automotive software development as well.

The software development must consider deep integration between hardware and software, strong focus development processes, strong supplier involvement, and safety-critical functionality. In this context, it is challenging to develop and distribute high-quality software at a high pace. Furthermore, the amount of software in the car increased exponentially the last decades [2]. To handle the increased complexity and the high amount of variations, a Software Product Line (SPL) was and is used. The SPL helps to handle changes, coordinates the software development worldwide and increases the software quality by reuse. However, increasing market pressure and fast changing requirements are challenging the existing solution with SPL. Agile Software Development (ASD) methods are a promising solution to keep with the market. The combination of ASD and SPL development is assumed to be difficult [3]. Agile development uses short development cycles and only a few Product Owners (PO), whereas SPL comprises several POs and a scoping process that hinders short iterations. This publication presents expected benefits and challenges for the integration of ASD into existing automotive SPL development. In summary, our contributions are as follows:

- Identify the expected benefits for the combination of ASD and SPL in the automotive domain.

---

[1] www.bundesregierung.de/Webs/Breg/DE/Themen/Energiewende/Mobilitaet/pod-cast/_node.html.

[2] http://www.elektronikpraxis.vogel.de/elektromobil/articles/608627/.

– Analyze the real-life challenges on Agile Software Product Lines in the automotive domain.

2   Related Work

To investigate the topic and find all the related work, we conducted a literature review to search for the Common Ground of Automotive Agile Software Product Lines [4]. The literature review revealed that there is no approach specifically tailored to the automotive domain handling the combination of ASD and SPLs. As presented in the study, it is therefore necessary to take the related research areas into account. Three major challenges for related research areas are identified, like the competitive pressure and the need to shorten development cycles (1) [5–9], different development cycle-times for related development systems (2) [9–11], and unclear management of software reuse and agile development (3) [5, 10, 12].

3   Study Approach

The goal of this study is to gain a better understanding of the expected benefits for a combination of ASD and SPLs. Furthermore, it identifies the automotive specific challenges that prevent the adoption of ASD within existing SPLs.

3.1   Research Questions

RQ 01: What are the expected benefits for the combination of ASD and SPL in the automotive domain?
RQ 02: Which real-life challenges hinder the adoption of Agile Software Product Lines in the automotive domain?

3.2   Research Design

The research reported in this paper builds on on-going research in collaboration with Daimler AG[3]. This study is based on a qualitative survey we conducted 2016 [13], the outcome of an internal expert workshop on Automotive Agile Software Product Lines with 40 participants as well as a discussion round on the ESE Congress 2016[4] in Stuttgart with 60 participants.

The interview study [13] took place between May and June 2016. The interviews were designed as exploratory semi-structured interviews to gain insights into the examined topic. For the semi-structured interviews an interview guide was implemented and tested in a pilot interview. We held an internal workshop with automotive experts at the Daimler AG. We let the participants vote on different statements to confirm the findings from the interview study. Furthermore, we presented these findings on the ESE Congress and repeated the voting.

---

[3] www.daimler.com.
[4] https://www.ese-kongress.de/.

3.3 Data Collection and Analysis

Research Sites and Participants. The interview participants are employees of an OEM and one automotive consultant. The interviewee selection was based on two criteria: First, the interviewee should have a work experience of at least two 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 with intent on implementing a software reuse strategy or vice versa. 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 from May to June 2016. In an internal workshop, 40 participants discussed the challenges in adopting agile practices to existing development processes. All participants were explicitly invited and from the automotive domain or related research institutes from universities. Furthermore, the workshop searched for requirements for the combination of ASD and SPL. The findings from the workshop were verified on the ESE Congress 2016 in Stuttgart, within a discussion round in the automotive agile session.

Interviews. The interviews consist of 14 face-to-face interviews and one group interview with two participants. 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. The analysis used the coding concepts of Straussian Grounded Theory, based on the classification of Stol et al. [14]. We used the three coding phases of Straussian Grounded Theory: open coding, axial coding, and selective coding [14]. The interpretive process of open coding breaks down the data analytically and generates categories and concepts. The concepts were grouped together and related to their subcategories in the axial coding. In the selective coding the central categories were defined.

3.4 Threat to Validity

This section treats the identified threats to the validity.

Interview and Interview Guide. The possibility of misunderstandings between interviewees and the researcher is a threat to validity. To minimize the threat, 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.

Research Sites and Participants. It is possible that the selection of the participants biased the outcome. For the internal workshop, it could be that the identified challenges are only in-house challenges. We validated the outcome on the ESE Congress 2016 to be sure that the challenges are not only in-house.

Researcher Bias. Data extraction and coding was done by Researcher 1. This could introduce bias due to misunderstanding and misinterpretation on the researcher side. To minimize the risk, the interviews were recorded and transcribed.

4   Results

The study reveals that the combination of ASD and SPL is a desired way of managing the software development in the future. There are several aspects why a combination is seen as beneficial.

4.1   Research Question 1

The study reveals two areas in which the participants identified a benefit.

Customer Collaboration. All participants mentioned that it is required for future success, to react on customer expectations faster. This ensures that customer-oriented products or features can be rapidly launched in the market with profit. One participant mentioned that with a rapid customer feedback the software development will be more effective. Not accepted solutions by the customer are identified in an early stage and could be dropped before they are further developed. The development capacity is better and more efficiently used. Trends in the customers behavior could be recognized and developed towards customer satisfaction.

Improvement of Development. The main expected benefit is an improved software development process. The development process benefits from several improvements, which could be categorized as: Transparency (1), collaboration within the development team (2), efficiency in development (3), flexibility (4), software quality (5), development speed (6), cost of delay (7), and a better verification and reuse strategy (8).

With the introduction of agile development, an agile mindset is presumed to be introduced as well. Most interviewees mentioned that transparency (1) in work will be resulting from the new way of collaborative working (2). With the consequential distribution of knowledge within the software development team, it is expected to be an open and genuine cooperation between employees (2). The developers mentioned that work could be more effective (3) by granting more responsibility to lower hierarchy levels. This results in less coordination for management approval. A possible solution is the resolving of too many levels in the hierarchy into flat hierarchies with self-organizing teams.

However, some participants are still critical about this proposed way of working. They mentioned that some developers do not want to change their working behavior and it is not possible to force a mindset change. Furthermore, the change need time and is not always necessary. A mixture of employees specialized in a specific field, and more general employees is necessary. With the right mixture, the development could benefit from in deep knowledge and flexibility (4) in the development. This flexibility is explicitly mentioned by the managers as a need to react on customer needs. For developers, the customer satisfaction is of secondary importance compared to software quality (5). They assumed that with a combination of ASD and SPLs it will be possible to deliver high quality software (5) at the required faster pace (6), due to increased software reuse and shorter release cycles. They further emphasized that it is important to

bundle the competence in-house, to deliver software faster and react on changing requirements. With faster in-house communication channels, one participant mentioned that this will be beneficial considering the cost of development and cost of delay (7). The developers mentioned that a good reuse strategy and SPL (8) is necessary to use parts of the software more often and save further money. All participants emphasized that an agile development speeds up the development to get a high quality software, whereas the existing SPL helps to ensure that already verified software is reused within a mature reuse strategy.

4.2   Research Question 2

Real-life challenges are evaluated, in order to combine ASD and SPLs in the automotive domain to handle agile development and software product lines. We focus on the development process, as the process must change and adopted to verify a smooth development in an agile way. The challenges could be categorized into organizational challenges (1), worldwide development (2), management (3), dependencies (4), synchronization processes (5), validation (6), release (7), and the software development process itself (8).

Organization. All participants mentioned that coordination is a challenge for introducing agile elements to the existing SPL. The existing hierarchy is changing slowly towards an open minded agile development. However, the existing processes are not bad and still valid. Special milestones in the processes verify worldwide coordination and planning, like scoping, to decide which features are relevant for implementation. By introducing agile development into the processes, it is seen that the coordination of the software development hinders a faster pace. The pace is influenced by coordination among the SPL. One challenge is to prevent a decomposition of the SPL. With many software variants and a faster development pace, it is the risk that the common software part becomes stunted. Therefore, it is important to consider synchronization points between software variants, several development processes like hardware and software, as well as supplier and in-house development.

Worldwide Distribution of the Development Team. A major challenge is the collaboration with suppliers and a worldwide distribution of the development team. Different cultures and different mindset are likely hindering an agile development. In addition, the purchase department is often interfering an agile collaboration with suppliers. This challenges are important to consider, while setting up the development team. Furthermore, the worldwide distribution of the development team leads to challenges in the team communication. Different time-zones, no face-to-face conversation and mistakes in translation represent big dangers. Furthermore, the process to maintain and scope the common software parts requires a lot of communication between all participants. This is seen as slowing down the development because of slow communication channels. A lot of planning and coordination is necessary to maintain the SPL. A participant mentioned that it is necessary that the communication is not only top-down, but in both directions.

Management. It is mentioned that the management does not want to give up any responsibility. With less responsibility on management level, scheduling of the development and reporting will be challenging. It is unclear for the managers how the agile software product line could be planned and features are scoped. Planning although is of high importance in an automotive development.

Dependencies and Synchronization. The automotive software development is a worldwide development with a lot of dependencies, like many included technical systems, test and verification steps and other developments domains like hardware and mechanical. A dynamic coordination is seen as necessity to introduce agile development practices into SPL development. The development process across several domains must be synchronized. Challenging is the fact, if just parts of the organization are working in an agile way and others not. Interfaces between those departments must be well organized and set up.

Validation and Release. Maintaining the compliance to standards while developing a lot of different software variants is seen as highly challenging. On the one hand, the development speed is increased but on the other hand, it is necessary to validate the software to maintain standards like ISO 26262 and other restrictions given by the law. It is a challenge to scale the test framework to test all variants within the SPL. It is unclear how far the automation of test could help in the process. Testing strategy must be context specific and scalable. With the use of the SPL already validated software parts could be reused. One participant mentioned the use of composable certification to be always compliant in all software variants. This form of certification is far from being legally allowed. New ways of certification and releasing software must be considered to maintain the development pace.

Software Development. One major challenge is the software development itself. Here is a lot potential for improvement. The identified challenges in the software development are clustered as technical issues (1), costs (2), requirements management (3), software architecture (4), software quality (5), safety regulations (6) and the use of SPL and variants (7).

As mentioned by the participants, the automotive domain is a cost-driven business. Therefore often the smallest possible hardware is selected to meet the requirements whereas this does not mean that there is a reduction of quality or functionality. But it leads to the unpleasant effect that in some cases, different variants of software are compiled separately to fit on the hardware. To get such a high modularity, it is necessary that the architecture is well chosen. One challenge is now, to maintain a good shaped architecture but have the possibility to integrate not foreseen features into the software. Furthermore it is necessary that all adoptions are always in relation to the selected hardware target. The developers mentioned that a benefit of the agile development is to re-prioritize features. The downside of this is that it is challenging to freeze the functionality for different variants of the SPL, because of late or incomplete requirements. The developers mentioned that it is important to analyze and prioritize the features to prevent chaos. This is even more important when developing within a SPL.

Scoping must always be considered to check for the affected variants. All variants must be validated to work as expected and defined by standards. The challenges in the software development are hard to tackle, but worth to consider.

5   Discussion and Conclusion

The identified challenges in the related work, such as the difficult management of software reuse and agile development [5,10,12] are valid for the automotive domain as well. A major finding of the study is that the use of agile elements in combination with software product lines require a precise analysis of dependencies to surrounding processes and organizational structures. For example, dependencies between departments and suppliers must be taken into account. It is also important to consider processes that are necessary to meet legal requirements. In the automotive sector global coordination is taking place due to global development. The currently typical form of coordination across several hierarchical levels (for example, to identify common components) is considered too slow and difficult to combine with the agile mindset. Dependencies between departments and compliance with rigid development processes are seen as an obstacle. It was important to all participants that the advantages of the software product line cannot be replaced by a more agile development. Particularly with regard to legal requirements and the verification and certification of software components, it is necessary to reuse software parts. This is particularly important in the automotive domain, as legal requirements require long-term certifications.

For future work, we plan to create and evaluate an Agile Software Product Line Automotive - Model (ASPLA-Model) for the adoption of ASD in the context of existing automotive SPL development. Next step will be deriving requirements for the ASPLA-Model based on the challenges reported herein.

References

1. Samuelsen, S.: The automotive future belongs to fuel cells range, adaptability, and refueling time will ultimately put hydrogen fuel cells ahead of batteries. IEEE Spectr. **54**(2), 38–43 (2017)
2. Broy, M., Krüger, I.H., Pretschner, A., Salzmann, C.: Engineering automotive software. Proc. IEEE **95**(2), 356–373 (2007)
3. Pohjalainen, P.: Bottom-up modeling for a software product line: an experience report on agile modeling of governmental mobile networks. In: Proceedings of 15th SPLC, pp. 323–332 (2011)
4. Hohl, P., Ghofrani, J., Münch, J., Stupperich, M., Schneider, K.: Searching for common ground: Existing literature on automotive agile software product lines. In: Proceedings of ICSSP 2017 (2017)
5. Babar, M.A., Ihme, T., Pikkarainen, M.: An industrial case of exploiting product line architecturesin agile software development. In: Proceedings of 13th SPLC (2009)
6. Farahani, F., Ramsin, R.: Methodologies for agile product line engineering: a survey and evaluation. In: Conference on 13th SOMET (2014)

7. Noor, M.A., Rabiser, R., Grünbacher, P.: Agile product line planning: a collaborative approach and a case study. J. Syst. Softw. **81**(6), 68–882 (2008)
8. O'Leary, P., McCaffery, F., Thiel, S., Richardson, I.: An agile process model for product derivation in software product line engineering. J. Softw. Evol. Process **24**(5), 561–571 (2012)
9. Olsson, H.H., Bosch, J., Alahyari, H.: Towards R&D as innovation experiment systems: a framework for moving beyond agile software development. In: Proceedings of IASTED (2013)
10. Díaz, J., Pérez, J., Alarcón, P.P., Garbajosa, J.: Agile product line engineering-a systematic literature review. Softw. Pract. Exp. **41**(8), 921–941 (2011)
11. Eklund, U., Olsson, H.H., Strøm, N.J.: Industrial challenges of scaling agile in mass-produced embedded systems. In: Dingsøyr, T., Moe, N.B., Tonelli, R., Counsell, S., Gencel, C., Petersen, K. (eds.) XP 2014. LNBIP, vol. 199, pp. 30–42. Springer, Cham (2014). doi:10.1007/978-3-319-14358-3_4
12. Ghanam, Y., Maurer, F.: Extreme product line engineering: managing variability and traceability via executable specifications. In: Agile 2009 (2009)
13. Hohl, P., Münch, J., Schneider, K., Stupperich, M.: Forces that prevent agile adoption in the automotive domain. In: Abrahamsson, P., Jedlitschka, A., Nguyen Duc, A., Felderer, M., Amasaki, S., Mikkonen, T. (eds.) PROFES 2016. LNCS, vol. 10027, pp. 468–476. Springer, Cham (2016). doi:10.1007/978-3-319-49094-6_32
14. Stol, K.-J., Ralph, P., Fitzgerald, B.: Grounded theory in software engineering research. In: Proceedings of 38th ICSE (2016)