An Empirical Study on the Current and Future Challenges of Automotive Software Release and Configuration Management

Houssem Guissouma*, Heiko Klare†, Eric Sax* and Erik Burger†
*Institute for Information Processing Technologies, Karlsruhe Institute of Technology, Germany
†Institute for Program Structures and Data Organization, Karlsruhe Institute of Technology, Germany
{houssem.guissouma | heiko.klare | eric.sax | erik.burger}@kit.edu

***Abstract*—Current automotive trends, such as autonomous and connected driving, are mainly enabled by embedded software that is deployed on a network of several, often more than one hundred, electronic control units. These software parts are responsible for many complex tasks concerning safety, comfort, energy management, and vehicle dynamics. Currently, they are deployed to the units at the end of the assembly line. Although a new software baseline of electronic control units is released in regular terms, normally six months, updates during after-sales are mostly conducted only in urgent cases, such as recall campaigns. Upcoming over-the-air services will enable more frequent updates to fix bugs and add new functionality. The possible alternatives of engines, chassis, and customer wishes lead to a high number of existing vehicle variants, so that the management of releases and configurations becomes more complex and costly.**

In a survey, we asked participants from different automotive institutions about the current state of practice, and the challenges they face during release development and management. This paper presents and discusses the main results of this survey: We have identified that field updates will be deployed more frequently in the future, and that over-the-air communication is an efficient way to realize them. The reported main update reasons are bug fixes and function improvement. However, the shortening release and update cycles, the increasing number of variants, and the multidisciplinarity in the automotive field are major challenges requiring suitable processes, methods, and tools to achieve software that operates correctly.

***Index Terms*—software maintenance, system validation, automotive engineering, configuration management, multidimensional systems**

I. Introduction

Since the introduction of the first electronic ignition systems in cars in the 1970s, the amount of electronics in vehicles has constantly been increased, driven by the growing number of electronic components on an integrated circuit, formulated in Moore's law [15]. Different mechanical function realisations have been replaced by *Electronic Control Units (ECUs)*, and new modules for more comfort and safety have been progressively added to vehicles. Nowadays, the on-board network of electronic systems together with the required power supply and management build the modern *electric/electronic (E/E) architecture*.

Due to this rising number of parts of E/E architectures and the increasing number of distributed software components inside a vehicle, model-based development has been adopted by most suppliers and *original equipment manufacturers (OEMs)* for the development of ECUs and software. It makes efficient and early testing of the models possible within the development process, leading to cost savings. During the different development phases, however, a high number of tools is involved that handle various views and abstraction levels. No single integrated design environment is capable of handling all requirements for designing and verifying the embedded system [11]. This leads to modelling consistency problems, especially when dealing with variant-rich systems and different life cycles. As a consequence, testing and validation plays an important role to ensure the required quality of the embedded hardware and software, and the safety of occupants. For that purpose, test processes and methods for automotive embedded system development have been developed [13].

Nowadays, most innovations in the automotive domain are realized in software. Within 30 years, the complexity of software that is deployed on a single car has increased from zero to more than ten million lines of code. Up to 40 % of the production costs of a car are attributed to electronics and software [6]. A recent study in a car manufacturing company has also exposed the still ongoing increase of software usage and complexity [1]. This increase of software complexity in vehicles leads to many challenges concerning the design, the quality assurance and the maintenance, which are for example discussed by Broy [6]. With the recent introduction of *software over-the-air updates (SOTA)* for infotainment and navigation systems, and the wish of OEMs to expand the application of this update technology to more safety-critical functions and to whole product lines, the issues of release and configuration management get even more accentuated.

We have conducted a survey to identify whether the above mentioned development poses a real challenge in the automotive industry. The survey is composed of various questions about update and release management, SOTA, current and future consistency issues, as well as the flexibility in exploring a multi-domain design space, e.g. of both hardware and software. Different companies and institutions from the German automotive industry participated in it. The results show that updates in the field, especially the SOTA case, will play a major role in the mobility of the future. However, their expansion is accompanied by critical consistency challenges arising mainly from the high number of system variants and the wide multidisciplinarity involved in the development teams. We have presented the initial results of this in a technical report [14].

The remainder of this paper is structured as follows: Section II presents an overview on foundations and state-of-the-art. The conducted survey is introduced in Section III, followed by a discussion of the results in Section IV. After giving an overview on related work in Section V, we conclude the report with a summary and future work in Section VI.

II. Foundations

In this chapter, main background knowledge about software release and configuration management with special focus on SOTA updates is summarized.

A. Software Updates

1) Software Release Management in Engineering: Releases in the automotive domain are regularly implemented and flashed at the end of the assembly line or during workshop visits. Individual changes are implemented separately and collected during a phase of normally six months. These changes are documented and tracked inside a so called *codeline diagram*, exemplarily depicted in Figure 1, describing the main development branch with the evolution of system releases.

Between the regular release phases, intermediate releases are mostly only developed for fixing severe bugs. In all other cases, rigidity of the release schedule is compulsory, otherwise the organization risks to lose the benefits of a common baseline. Updates, i.e. flashing during usage in the field, for safety-critical functions are primarily only provided to fix severe bugs because of the required effort and induced public interest.

2) Traditional Software Updates: Automotive software updates are traditionally done in workshops, mostly for fixing a severe bug. First, the driver is informed by mail about the availability of an update and is asked to bring the car to the next workshop. Depending on the bug severity and the kind of the visited workshop, the bug fix is sent from the OEM before or while the car is in the workshop. When the software update is ready, it is flashed on the target ECU through the *On-Board-Diagnose (OBD)* interface. Each ECU update can take about 15–90 minutes [12]. Depending on the safety criticality of the update, the technician may also need to test the flashed ECU through e.g. performing a diagnosis via OBD or functional tests. The described update process needs both the owner and technician to invest time and effort, which produces high costs and some customer's inconvenience.

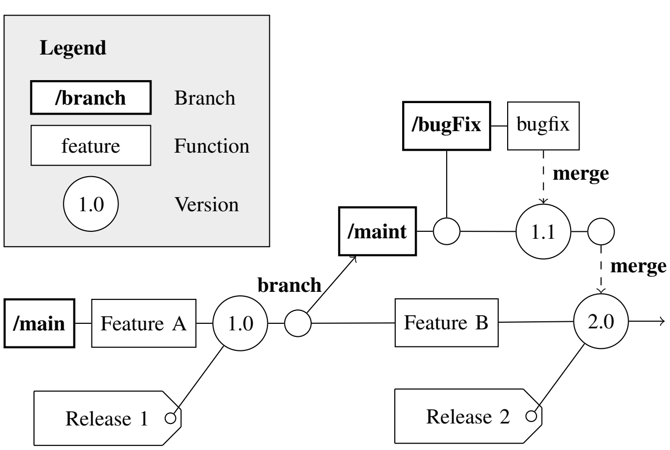

Figure 1: Exemplary codeline diagram representing one main-line development

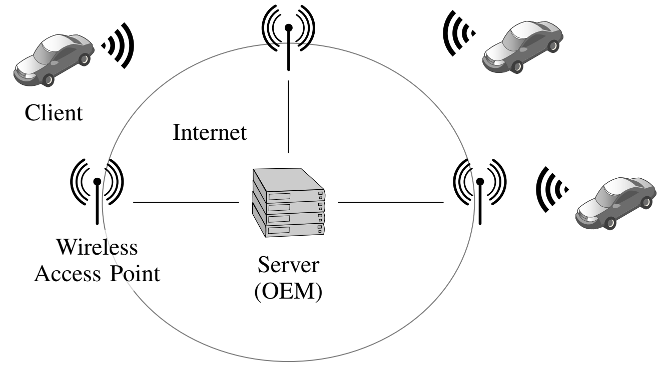

Figure 2: Client-server architecture of a SOTA network

3) Software over-the-Air Updates: A SOTA system is typically built according to a client-server architecture, as depicted in Figure 2. The server allows the OEM to deploy updates to the clients, which are the SOTA processing units of vehicles. A client connects to the server through a wireless access point, downloads the software update binaries and distributes them through the internal bus systems to the appropriate ECUs.

The first vehicles with an over the air updates capability are already in the market. The main use case is for less safety-critical systems like infotainment applications and navigation maps. Like the US car company Tesla, which offers updates for the autopilot remotely, many OEMs want to introduce SOTA also for safety-critical applications. Wireless bug fixes would enable important cost savings as reported by Bird et al. [4], who predict that the total worldwide OEM cost savings from SOTA updates are expected to reach \$35 billion in 2022. However, SOTA updates are accompanied by some important challenges. In addition to the security risks arising from opening the car to the outer environment with the threat of reprogramming its embedded systems, the high number of variants within automotive product lines makes it difficult to keep the consistency of the models and views involved in the development and validation of new releases.

B. Software Configuration Management

Software Configuration Management (SCM) summarizes the activities required for the management of the software parts of a product along its life cycle. It comprises configuration identification, configuration control, status accounting, review, build management, process management, and teamwork [3]. SCM activities play a crucial role for the product quality in the automotive industry and need to be continuously optimized in order to keep pace with the increasing system complexity and the shortening software life cycles.

When integrating changes in form of new version as described in Subsection II-A1 into the system, consistency checks of relevant configurations have to be performed. This can be done based on dependency descriptions, such as interfaces or contracts between composed configuration items [8]. Also, regression tests are needed before each release to make sure that the introduced changes did not lead to other still unrecognized system errors or failures.

Besides, and because of different country regulations, multiple equipment models and different construction and hardware/-software realization alternatives, the OEMs in the automotive industry must maintain large system variant spaces. Several methods and tools like PREEvision[1] already exist to manage E/E architecture variants during the development. Nevertheless, dealing with system variability with the shortening update cycles of software due to e.g. software bugs is still a challenge that we want to investigate in this work.

III. Conducted Survey

A. Research Method

We employed a questionnaire to collect answers on challenges, the current state-of-the-art and expected future trends in automotive software release management. We mainly used closed questions with a predefined answer set, but also provided open questions for additional concerns of the participants, especially for identifying future trends. As the potential answers, even if numerically coded, are not on an equidistant scale, we only consider the answers given on an ordinal scale. Therefore, we used established Likert-type scales [10] or derivations of it. Hence, we can sum answers across participants, compare selected answers and, thus, perform statistical tests on the results, but cannot, for example, compute means.

According to the classification scheme by Shaughnessy et al. [17], we performed a cross-sectional study, which gathered data at one specific point in time, not considering developments over time. Nevertheless, specific questions are used to identify an expected or predicted trend regarding specific topics. Our questions regarding the current state-of-the-art primarily aim at *facts*, while the questions regarding future trends also consider *opinions* of the participants. The selection of participants was quasi-random. It was not based on certain properties of the persons, but also not completely random from all persons of interest. Due to their restricted reachability, we had to employ open mailing lists and personal contacts to find participants.

[1]PREEvision is a widely-used tool of the company Vector Informatik for model-based development of E/E architectures in the automotive industry.

Category	Questions
Environment (field of work / product)	*2*
State-of-the-art update / variant management	*3*
Field updates and software over-the-air updates	*5*
Consistency of updates	*3*
Multi-domain development and design space	*5*
Conclusions regarding limitations and desires	*2*

Figure 3: Categories of survey questions

B. Design and implementation of the Survey

In order to design the survey, we classified the information to be collected in six categories, which are listed in Figure 3. We then developed 20 questions based on these categories starting from a survey on the state-of-the-art towards an identification of future trends and challenges. All the questions were first reviewed by an additional expert on empirical methods outside the automotive domain, before a pretest group of three pseudo-participants checked them for understandability.

As all participants were from German automotive industrial and research institutions, the questions have been written in German language. After finalizing and reviewing the questions, they were published in an online survey using the tool LimeSurvey[2]. Subsequently, the link to the survey was sent to different contacts from the automotive sector, including OEMs, suppliers (or tier-one companies), scientific institutions and licensing offices, in multiple mailing lists with a motivating invitation text.

C. Content of the Survey

The main goal of the survey was to identify the current state-of-the-art and the challenges of release and configuration management in the automotive industry. The focus is set on software updates in the field, especially the SOTA case, and the resulting challenges to maintain the consistency and continuously validate the product lines along their evolution. In addition, the available degree of freedom to realize functionalities in hardware or software and its effect on the product were inquired. These research subjects are treated in the 20 questions of the survey, which have been classified into the six categories depicted in Figure 3.

IV. Results and Discussion

In this chapter, we present and discuss the results of the conducted survey in order to identify current and future challenges, as well as expected future trends when dealing with updates and variant-rich systems in the automotive industry.

A. Participant's Environment

The survey was answered by 51 participants from different segments, most of them being engineers of OEMs and tier-one companies (see Figure 4). Most participants develop and

[2]https://www.limesurvey.org/, retrieved 2018-09-03

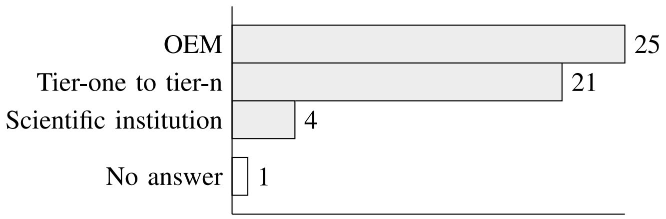

Figure 4: Work segments of participants

maintain software (57 %) and hardware (45 %) components. In addition, an important part of the participants is responsible for E/E systems (35 %) or E/E architectures (33 %). The sum of percentages for the answer alternatives exceeds 100 % because multiple answers were allowed as multiple systems or one heterogeneous system with different parts can be partially developed by the same person.

B. Identified Current Challenges

1) Variant Management: The number of variants managed by the participants, defined as the number of system/product derivations at the same time, is diverse, as shown in Figure 5. Most of the participants (35 %) maintain a small number of variants (less than 10), whereas 25 % are responsible for more than 50 system variants. Another 33 % manage between 10 and 50 variants. When managing more than one system, participants were asked to give an estimated mean value of the variants numbers of all managed systems.

2) Software Release Frequencies: New releases of automotive software during the development normally deal with corrections of bugs or system optimizations through changing the models or the requirements, and are usually unavoidable during each product development. The participants reported that already during the development of their products many new releases in form of new E/E states are created. 59 % and therefore the majority of the participants indicated a number of more than three releases per year, as it is shown in Figure 6.

In contrast, the frequency of releases of new E/E states for products in the field is more diverse and overall lower. About 65 % of the participants responded with a release frequency between less than once a year and twice a year, while 16 % indicated that their product gets a new release more than three

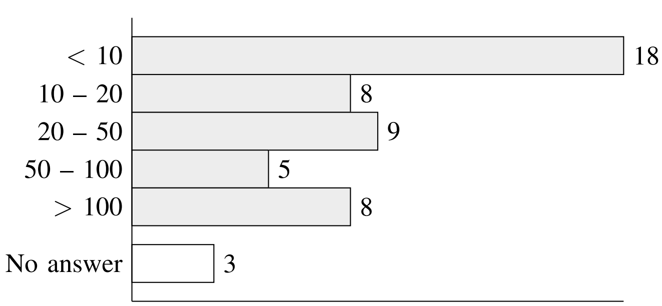

Figure 5: Number of variants managed by the participants

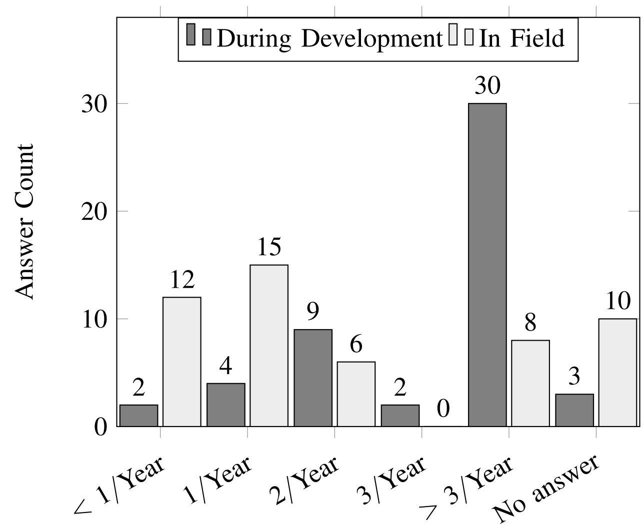

Figure 6: Frequency of system releases during development and for products in the field

times a year, as depicted by the light-colored bars in Figure 6. This shows that most automotive systems in the field still get no or seldom new releases. One potential reason for the different release frequencies is the dependence on the functionality of the component to be updated. For example, an update of the navigation system software will potentially be performed more frequently than an update of a more safety-critical component or even the hardware of a sensor. Nevertheless, this is only a hypothesis, which one would have to confirm by analyzing the correlation between type of the updated product and the release frequency.

3) Software Update Frequencies: In this survey, an update is defined as an installed new release or a change which has been added to the system in the field. According to most participants, the frequency of field updates is still very low and estimated at once a year or less as shown in Figure 7. Those reported frequencies are for the most part in concordance with those of after sales releases as depicted in Figure 6. In fact, 80 % of the participants who reported conducting field updates less than once a year also revealed that they develop at most one release annually. Additionally, 69 % of those who develop annual updates stated that releases are developed at most twice a year. Nevertheless, the update frequencies of products in the field are overall lower than those of releases. As a hypothesis for future studies, we formulate that the number of releases for products in the field is usually higher than the number of executed updates, as already indicated by this comparison.

The current time span for which field updates are offered is reported by 35 % of the participants to be relatively short and ranges between one and three years. This is probably a kind of after-sales guarantee for repairing all potential malfunctions or bugs directly by the OEM. Nevertheless, 20 % offer field updates for at least 7 years. As a hypothesis, introducing SOTA updates may increase this time span to the whole vehicle life cycle. Almost all participants (94 %) agreed that field updates

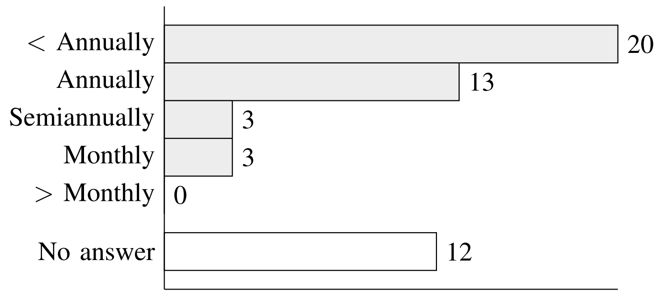

Figure 7: Frequency of updates in the field

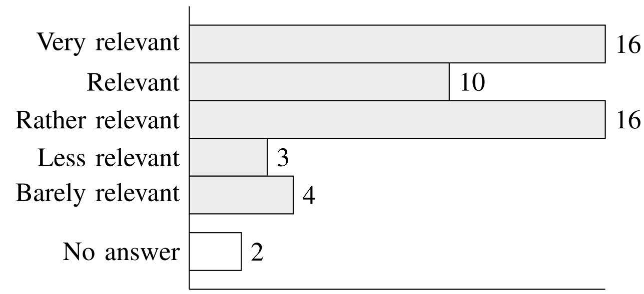

Figure 8: Importance of decision freedom for multi-domain function realization within automotive institutions

with short life cycles will gain more importance for seamless connected mobility.

The participants agreed that the main reasons for current updates are bug fixes (88 % agreement) and function improvement (67 % agreement). Costs optimization and component availability are, according to the questionnaire responses, less relevant reasons for an update than the ones mentioned above.

4) Consistency of Software Updates: In order to ensure compatibility of updates with the existing E/E environment, consistency checks are essential and determine the update quality and safety. 45 % of the participants reported that these checks are representing at least 50 % of the work effort needed to solve occurring inconsistencies. Compared to the validation of the initial product, the majority of the participants reported that the effort to validate an update is equal (29 %) or even less (33 %). Nevertheless, 28 % revealed that this effort is higher or much higher than within the initial validation process.

The participants were asked for the main reasons for inconsistencies of their products that occur in their development processes (see Table I). Most of them stated that a high number of variants is the most important reason for inconsistencies (71 %), followed by missing methods and methodologies (53 %) and the high interdisciplinarity in the development process. To discover the arising inconsistencies, Hardware-in-the-Loop testing is a commonly used method in the industry, as agreed by the majority of the survey participants (80 %).

Regarding SOTA updates, we presented three statements to which the participants could indicate their agreement, shown in the first half of Table II. Most of the participants (77 %) revealed that their institution is currently working on the introduction of SOTA updates with different grades of intensity. Also, the majority agreed that security is considered as a main concern for SOTA updates. However, answering the question whether license relevant updates, i.e. updates that need to conform with a national or international licensing norm or law, will represent an important part in this case, a stronger variance of the answers could be observed. Nevertheless, the majority agreed that the license relevant updates will play an important role in the future.

Reason	Agreement
High variants number	71 %
Missing methods	53 %
High multi-disciplinarity	53 %
High documentation efforts	47 %
High communcation efforts	43 %

Table I: Agreement with reasons for inconsistencies during product developmen (ordered by degree of agreement)

5) Multi-Domain System Design: During the development of complex automotive systems, the question whether to realize a function in software, hardware or mechanical parts is frequently posed, especially when going from the system requirements to system design. The decision is mainly taken after initially analyzing the design space and evaluating different possibilities. When considering only hardware and software, this methodology is known as hardware/software codesign. Allowing the engineers to distribute the function parts over an enhanced design space of hardware, software or mechanical components is an important factor for a successful design and an optimized product. Only around half of the participants reported that they have enough degrees of freedom to make this decision, but 82 % revealed that this freedom of decision is important for their institution, as shown in Figure 8. As expected, the majority of the survey participants agreed that a higher decision tolerance has a strong influence on the quality, the efficiency and the maintenance of the product, as shown in the second half of Table II.

C. Future Challenges

The answers to the survey indicate that the frequency of field updates will significantly increase in the upcoming five to ten years, as illustrated in Figure 9. Updates in short cycles of less than a year will become more important. Possible reasons for this predicted development are an increasing number of services and functions of vehicles that are based on software, and more synchronization needs due to distributed development groups. This can result in a more error-prone development making bug fixes during use necessary. Furthermore, the customer's wishes for innovations and individualization demand for software updates in shorter and longer terms. These demands have to be confirmed in a further study.

Besides, the reasons for the updates will partially change. While bug fixes and function improvement will continue to

	fully agree	agree	rather agree	rather disagree	disagree	fully disagree
Current state and future of SOTA updates in the automotive industry						
My institution currently addresses the introduction of SOTA intensively.	23.5	**27.5**	25.5	5.9	2.0	3.9
Security plays an important role in the introduction of SOTA.	**66.7**	17.6	3.9	0.0	0.0	2.0
License relevant updates will represent an important part of SOTA updates.	13.7	19.6	**21.6**	15.7	9.8	3.9
Expected influence on design decision for multi-domain function realization						
More degrees of freedom for design decisions lead to …						
…higher quality of the product.	—	19.6	**43.1**	21.6	2.0	—
…more efficient development of the product.	—	27.5	**43.1**	11.8	3.9	—
…more efficient maintenance of the product.	—	19.6	**52.9**	9.8	3.9	—

Table II: Participant's agreement with specific statements (in percent of all participants, highest value per statement boldfaced)

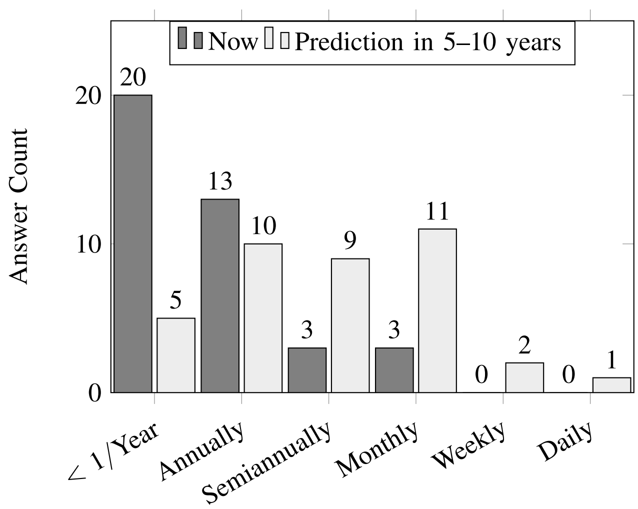

Figure 9: Predicted evolution of the update frequencies in the next 5 to 10 years

increase, the costs optimization as a reason for updates will play a greater role, as reported by the survey participants. 67 % of the participants agreed that bug fixes as a cause of field updates will increase in the upcoming five to ten years, and 86 % expect a growth of updates for the sake of function improvement. As to costs optimization updates, their growth is expected by approximately half of the participants, and only 8 % think that their importance will decrease.

D. Need for Consistency

The variety of vehicles of one type derives from different engines, chassis, gearboxes and regional specifics. Multiplied by several options for the individual equipment and the number of vehicle types of one OEM, the number of variants can exceed 10 million [16]. The number of variants reported by the participants, shown in Figure 5, is comparatively low, as the individual responsibility usually only concerns a specific part of the vehicle instead of the vehicle as a whole. Even if the network of more than 100 ECUs in one vehicle is neglected and the individual software parts are regarded as one cluster only, there is a tremendous need to check if additional software runs correctly in interaction with the existing one. Of course, not each update is relevant and has to be deployed on each vehicle of one brand, but for example in the so called "Dieselgate" more than 5 million cars, only counting those in Germany, are affected. 71 % of the survey participants wish that more processes, methods and tools support variants management, which, in the survey, is considered as the main challenge in the release and configuration management of modern and future automotive systems.

Effective and efficient consistency checks before performing software updates are one of the only chances to avoid software that is incompatible and causes potentially dangerous failures. 61 % of the participants would like to be supported by more processes, methods and tools performing consistency checks.

E. Threats to Validity

We finally discuss threats to the validity of our study and how we tried to handle them. The internal validity refers to the validity of the data acquisition instrument and process, in our case the questionnaire. The external validity is concerned with the generalizability of the results.

1) Internal Validity: The main threat to internal validity is that different participants had different understandings of the questions or answer options. To ensure that questions were understood uniquely and as intended, we used our pretest to check formulation and wording of the questions. In addition, the range of given answer options also helps to understand the meaning of the questions as intended by the authors. The unique understanding of the answer options was supported by clear indications of its values. So we defined terms like "often" by a value of frequency. Finally, an ambiguous question would usually result in a high variance of the answers to one question, which cannot be perceived in any of our results.

Another potential problem is the ordering of questions. A bad arrangement can result in questions that prime certain terms or concepts in the mind of the participant, which influence the answers to subsequent questions. Without such priming, questions would have most likely been answered differently. This effect is hard to avoid as a specific order of the questions finally has to be chosen, and priming through contemplating on previous questions is unavoidable at all. However, in our review and pretest we looked at such ordering effects and, to the best of our knowledge and imagination, could not identify potential negative effects of the question ordering.

Finally, participants may enlarge problems to support the researchers of the survey or to present themselves as very concerned or analytic thinkers. On the other hand, participants may diminish challenges for not admitting unsolved problems in the company. This bias is hard to control, but we assume that it is limited due to the given anonymity, which cannot make the participant expect any positive or negative consequences.

2) External Validity: Regarding external validity, we see in particular that all participants come from a small set of Southern German car manufacturers or tier-one companies. This clearly limits the generalizability to arbitrary other car manufacturing regions. However, at least in the area of classical combustion engine cars, Southern Germany's car industry is among the world-wide leading ones, hence we can reasonably assume that our findings can be transferred to other leading car manufacturers of such cars. Furthermore, although the respondents do only belong to Southern German companies, the number of these companies is higher than 15 and therefore represents a large variety of environments. However, it is also reasonable to assume that results for immature manufacturers or those of low quality cars may differ. The hypothesis that maturity of the companies also influences the applicability of our results is reasonable, as mature companies already managed the handling of many variants and the organization of cross-generational reuse of designs, as well as the exploitation of car platforms, which is all less developed in immature companies. However, all these properties of the design (handling of variants, cross-generational reuse and platforms) highly influence the answers on our questions. Therefore, we assume that our results are only applicable in regions, where car industries also use such mature design properties and approaches.

Another threat to the external validity of our study is that distributing the questionnaire by mailing lists and personal contacts may have restricted the set of participants to a specific subset of employees in the car industry, e.g., only those on a specific hierarchical level. We reacted to this threat by using different channels and also the contacts of different persons for distribution to reach a variety of people that represent an adequate sample of relevant groups of employees in the car industry. Furthermore, we designed the questions in a way that, to the best of our imagination, the answers do not depend on the position of a participant within its company, except from his or her working area, which was therefore covered by one of our questions (see Figure 4).

Finally, the number of respondents (51) represents only a small sample of the people that our survey addressed. Additionally to the restricted generalizability of our results to arbitrary car manufacturing regions, this could also reduce the generalizability for the Southern German industry, of which we selected the participants. There are two reasons why we do not consider this as a threat to the validity of our survey: (a) the participants belong to at least 15 different companies, which induces a variety of different environments that are covered by the survey; and (b) the variance of the answers to all questions is low, which is not only indicative for the unambiguity of the questions, but according to the law of large numbers also an indicator for a good sampling of participants.

V. Related Work

The rising complexity of automotive systems, driven by new trends and services in automated and connected driving, has been subject to different other empirical and technical studies. In [7], challenges in the design of automotive functions involving both high-level control models and their actual implementations on multiprocessor platforms have been discussed. A holistic cyber-physical systems design approach combining both layers is introduced as a necessary measure to manage development complexity, as identified in Subsection IV-B5.

Braun et al. [5] conducted guided interviews with experts of German OEMs, suppliers and research institutes to find significant trends and issues concerning automotive E/E architectures. In addition to power management systems, self-driving cars and vehicle-to-X communication, most experts consider SOTA updates to generate an extensible E/E architecture as one major future trend. However, the work reveals that those updates are still challenging and that software modularity is a necessary feature to reduce the update development effort, which conforms to the findings of our work.

The technical realization of SOTA updates, which are expected to improve the frequency of field updates significantly, has been topic of many recent research works. Steger et al. [18] proposed a generic framework enabling secure and efficient wireless software updates throughout a vehicle's lifetime. They use the IEEE 802.11s standard for realizing wireless networks that allow parallel updates of multiple vehicles, and include adequate methods to reduce security risks. In our study, we identified that the shortening release cycles together with the high number of variants are the main challenges that should be supported by efficient configuration management methods, tools and processes to allow for safe SOTA updates.

Similar challenges to those presented in this study can be observed in other cyber-physical system domains: one of them is the field of automated production systems. Li et al. [9] showed that the complexity of those interdisciplinary systems (mechanics, E/E and software) together with the different frequencies of the involved innovation cycles are the main challenges in the system and release development. An expert survey has been conducted to confirm this hypothesis and the results showed that paradigm changes are needed to handle this complexity. However, missing solutions for module management as well as the diversity of variants and versions were identified as main innovation inhibitors.

In this work, we showed that the main reasons for inconsistencies in automotive release development are the high variants number and the missing methods to handle them. These inconsistencies could occur between the different models and views involved in the model-based development of the system or sub-system. One of the most commonly used languages for modelling large-size object-oriented systems is the *Unified*

Modeling Language (UML). Bashir et al. [2] present the state-of-the-art of formal and non-formal techniques to manage inconsistencies between different UML model types, such as class, sequence and state chart, during software maintenance. They found that formal techniques, although still not established in the application fields, are one main topic of research studies. However, according to this work, some kinds of inconsistencies like vertical (e.g., inter-model) and semantic consistency have not been explored in depth yet.

VI. Conclusions and Future Work

The development of functionality such as automated and connected driving in automotive systems amplifies the demand for software realizations. With the increase of software in size, complexity, and versions due to the growing number of vehicle variants and software releases, ensuring proper operation becomes more difficult. Additionally, customers demand more functions that are driven by external technologies, such as smartphone integration. This development requires automotive software to be easily updatable after initially deploying it into the car, especially with software over-the-air updates.

We have conducted a survey in the German automotive industry to get an overview on the state-of-the-art in update and variant management for vehicles, the current challenges in introducing software over-the-air updates, and the assurance of their consistency with the rest of the vehicle system. An important portion of the participants declare a high number of car variants to be handled and state that updates in the field are currently difficult and therefore rarely performed. To achieve properly operating updates, they demand for new ways in their validation before sending them into the field. Validating consistency with the large number of existing software releases is essential to avoid inoperable cars and severe accidents.

To handle this demand for consistency, new approaches for the development of vehicle systems have to be developed. In our survey, we have found that methods such as hardware-in-the-loop are already established for testing updates. Nevertheless, most participants agree with the statement that new methods are needed that support the development process in earlier phases by exploring domain-spanning design spaces, e.g., allowing more decision freedom in the hardware/software function distribution, and to support consistency management. This yields a need for appropriate solutions and tools and expresses a motivation for their development in future work.

In a future survey, the gathered data should be analyzed for interesting correlations. An example for such a potential correlation is the dependency between the numbers of managed variants and the effort for ensuring and validating consistency. Furthermore, some correlations discussed in this paper were only speculative, since the required data had not been collected, and should thus be verified. One example is the relation between update frequencies and the product type, especially regarding its safety-criticality. Together with a trend for the relation between safety-critical and non-safety-critical components in cars, this would yield a prognosis for update frequencies for products in the field.

References

[1] V. Antinyan et al. *Monitoring evolution of code complexity in agile/lean software development - a case study at two companies*. In *13th Symposium on Programming Languages and Software Tools, SPLST 2013*, 2013.

[2] R. S. Bashir et al. “UML models consistency management: Guidelines for software quality manager”. *International Journal of Information Management*, 36(6), 2016.

[3] S. P. Berczuk and B. Appleton. *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*. Addison-Wesley, 2002.

[4] C. Bird and E. Juliussen. *Improving software, reliability & innovation*. Technical report, IHS Technology, 2015.

[5] L. Braun, M. Armbruster, and E. Sax. *Stakeholder issues concerning the automotive E/E-architecture*. In *2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles International Transportation Electrification Conference (ESARS-ITEC)*, 2016.

[6] M. Broy. *Challenges in automotive software engineering*. In *Proceedings of the 28th International Conference on Software Engineering*, ICSE ’06, 2006.

[7] D. Goswami et al. *Challenges in automotive cyber-physical systems design*. In *2012 International Conference on Embedded Computer Systems (SAMOS)*, 2012.

[8] C. Heinisch, V. Feil, and M. Simons Daimlerchrysler. “Efficient configuration management of automotive software”, 2004.

[9] F. Li et al. *Specification of the requirements to support information technology-cycles in the machine and plant manufacturing industry*. In *14th IFAC Symposium on Information Control Problems in Manufacturing*, 2012.

[10] R. Likert. “A technique for the measurement of attitudes.” *Archives of Psychology*, 140, 1932.

[11] K. D. Muller-Glaser et al. “Multiparadigm modeling in embedded systems design”. *IEEE Transactions on Control Systems Technology*, 12(2), 2004.

[12] H. A. Odat and S. Ganesan. *Firmware over the air for automotive, fotamotive*. In *IEEE International Conference on Electro/Information Technology*, 2014.

[13] E. Sax. *Automatisiertes Testen Eingebetteter Systeme in der Automobilindustrie*. E. Sax, editor. Carl Hanser Verlag GmbH & Co. KG, 2008.

[14] E. Sax et al. *A Survey on the State and Future of Automotive Software Release and Configuration Management*. Technical report, Karlsruhe Institute of Technology, Department of Informatics, 2017.

[15] R. R. Schaller. “Moore’s law: past, present and future”. *IEEE Spectrum*, 34(6), 1997.

[16] H. Schleich, J. Schaffer, and L. F. Scavarda. *Managing complexity in automotive production*. In *19th international conference on production research*, volume 100, 2007.

[17] J. J. Shaughnessy and E. B. Zechmeister. *Research methods in psychology*. Alfred A. Knopf, 1985.

[18] M. Steger et al. *Generic framework enabling secure and efficient automotive wireless sw updates*. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016.