Karlsruhe Reports in Informatics 2017,11
Edited by Karlsruhe Institute of Technology,
Faculty of Informatics
ISSN 2190-4782

A Survey on the State and Future of Automotive Software Release and Configuration Management

Eric Sax, Ralf Reussner, Houssemeddine Guissouma and Heiko Klare

2017

KIT – University of the State of Baden-Wuerttemberg and National Research Center of the Helmholtz Association

Fakultät für Informatik

A Survey on the State and Future of Automotive Software Release and Configuration Management

Eric Sax, Ralf Reussner, Houssemeddine Guissouma, Heiko Klare

Abstract

In modern vehicles, embedded software is mainly the driving force for the realisation of new functionality. Spread on more than hundred electronic control units, software parts work together in a network to fulfil certain tasks of safety, comfort, energy management and of course vehicle dynamics. Currently, this software is flashed at the end of the assembly line. In regular terms (normally six months), a software baseline of all the electronic control units is released. As a consequence, a distinct variety of releases is deployed on vehicles on the road for each vehicle type. Combined with the alternatives of engines, chassis and customer wishes, the variants of vehicles rise to a number beyond millions. Opening now the chance for over the air updates demands for restrictive and rigorous consistency checks before any update is spread among all the vehicles of one brand or type in the field.

In an empirical study based on a survey, we collected and interpreted the answers of participants from different automotive institutions concerning the current state of practice and the faced challenges during release development and management. The outcome of the survey revealed that field updates are getting more and more important due to the rising software part inside vehicles and that over the air communication is an efficient way to realise them in the future. The shortening release and update cycles together with increasing number of variants and the multi-disciplinarity in the automotive field were identified as the main challenges in the current development of automotive engineering.

Keywords: software over the air updates (SOTA), automotive, consistency checks, release management, variant management, multi-domain development

1 Introduction

The number of electric and electronic components in cars is constantly increasing. New functions for more comfort and safety are realised by embedded electronic systems executing software algorithms, which process sensors signals and control different actuators. One of the most important enablers for this rapid development of *electronic control units (ECUs)* is the invention of bus communication systems, starting with the *Control Area Network (CAN)*, which was initiated by the German supplier company Bosch in the 1980s. These systems allow the different ECUs to communicate with each other and exchange function relevant signals. This led to a complex so called *electric / electronic (EE) architecture*, which refers to the on-board network of all sensors, actuators, ECUs, as well as power supply and management within vehicles.

Due to this rising number of parts of EE 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. This allows an efficient and early testing of the models, using for example rapid prototyping, within the development process. In that way, errors can be discovered and corrected in early development phases leading to important cost savings. During the different development phases, however, a high number of tools handling various views and abstraction levels are involved and no single integrated design environment is capable of handling all requirements for designing and verifying the embedded system [Mul+04]. Since the information processed by different tools is not completely independent, this leads to 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 drivers and passengers. The state-of-the-art of processes and methods for testing automotive embedded systems was summarised by Sax [Sax08].

Nowadays, most innovations in the automotive field are realised in software. Within 30 years, the complexity of software that is deployed on a single car increased from zero to more than ten million lines of code, and up to 40 % of the production costs of a car are attributed to electronics and software [Bro06]. A recent study in two companies, of which one is a car manufacturer, has also exposed the still ongoing increase of software usage and software complexity [Ant+13]. Due to this dramatic increase of the size of software in vehicles, many challenges concerning the design, the quality assurance and the maintenance, which are for example discussed by Broy [Bro06], arose. With the recent introduction of *software over the air updates (SOTA)* for in-car information, entertainment (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.

For this reason, we implemented a survey to identify if 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. hardware and software, for selecting problem solutions. Different companies and institutions from the German automotive industry participated in it. The results showed 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. These should thus be subject to a deep research and require new methods and tools.

The remainder of this report is structured as follows. Section 2 presents an overview on foundations and state-of-the-art. The conducted survey is introduced in Section 3, followed by a discussion of the results in Section 4. We conclude the report with an overview of future work in Section 5.

2 Background and State-of-the-Art

In this chapter, main background knowledge in the fields of automotive electric/ electronic development as well as software release and configuration management with special focus on SOTA updates are presented. Furthermore, some results of a couple of related empirical studies are summarised.

2.1 Electric/Electronic Development in the Automotive Industry

The increase in performance, robustness and availability of semiconductor technology enables the integration of new functionality in all areas of daily life. Especially mobility systems profit from this trend. New software applications for automated driving, electrical power train management and new business services are on the rise in modern vehicles. Consequently, it is necessary to integrate agile and iterative software development methods into traditional, mechanically driven engineering processes, which are rather conservative and lengthy. This introduction of new development processes is one of the main current challenges in the automotive industry. Tools to support such agile processes are rapid control prototyping, model-based design and automated code generation.

The hardware platforms integrated in modern cars are a network of more than 100 ECUs in six to eight subsets of domain clusters, which represent different functional, areas such as power train, body or infotainment and telematics. These clusters are combined by one gateway. The trend leads towards one ECU for each domain to bundle functionality of high communication and serve as decentralised gateways.

2.2 Software Updates

2.2.1 Software Release Management in Engineering

In the automotive domain, it is current state of the practice that software is individually implemented in certain states of maturity, according to the established A to D pattern, and flashed at the end of the assembly line in common releases. This is regardless of the software development method, agile or conservative or otherwise.

During a phase of normally six months, changes in individual functionality or enhancements are separately implemented and collected. This is done reliably by sophisticated change management processes. After that period, a new baseline is available for the flash process in production. Intermediate releases are only possible for fixing severe bugs ("hot fixes"); flashing even during the usage in the field is only foreseen to fix dramatic bugs, because it causes a lot of effort and public interest. In all other cases, rigidity of the release schedule is compulsory, otherwise the organisation risks to lose the benefits of a common baseline.

2.2.2 Traditional Software Updates

Traditionally, a software update requires the owner to bring his or her car to a workshop, in which a technician flashes a new software version on the ECU. This requires both the car owner and a technician to invest time in the software update, which produces corresponding costs.

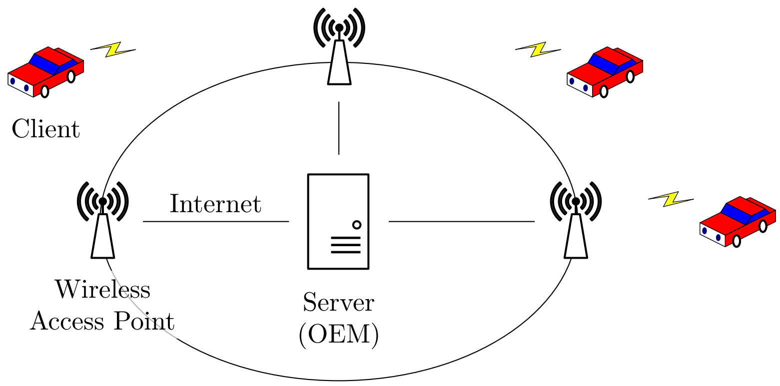

Figure 1: Client-server architecture of a SOTA network

Today, the main reason for performing a software update after sales is a severe bug. In this worst case, the driver is informed by regular mail that there is a software bug within one component and that he must bring the car to the next car workshop. Depending on the bug severity and the kind of the visited workshop, the adequate bug fix is ordered from the OEM before or while the car is in the workshop. When the software update is ready, which can take several days, it is flashed on the concerned ECU over the *On-Board-Diagnose (OBD)* interface using cables. Each ECU update can take about 15–90 minutes [OG14]. OBD is an open self-diagnostic capability of vehicles monitoring all exhaust gas relevant signals and other important control signals, and is an obligatory feature of all existing cars with combustion engines in Europe since 2004. Depending on how safety critical the update is, the technician may also have to test the flashed ECU, for example by performing a diagnosis via OBD or functional tests. Only then, the owner is informed that the repaired vehicle is ready for pick-up.

If the bug is caused by a hardware failure, the update usually consists of exchanging and re-testing the hardware module.

2.2.3 Software over the Air Updates

Executing updates over the air is a quite new feature of modern vehicles, which can enable immense cost savings by avoiding the expensive traditional update process for bug fixes as described in Subsection 2.2.2. According to Bird et al. [BJ15], the total worldwide OEM cost savings from SOTA updates are expected to reach $35 billion in 2022.

While different OEMs integrated the SOTA capability for less safety-critical systems like navigation maps and infotainment applications, the US car company Tesla remains the only OEM to integrate SOTA for core ECUs allowing for example to update the autopilot function remotely.

The most established structure of a SOTA update system is the client-server architecture. Here, the OEM has the task to manage its fleet and run a server that provides an update service. The car has to be equipped with a SOTA processing unit including the client function, which connects to the server via a wireless access point and downloads the required software update binaries from the OEM server. This structure is depicted in Figure 1.

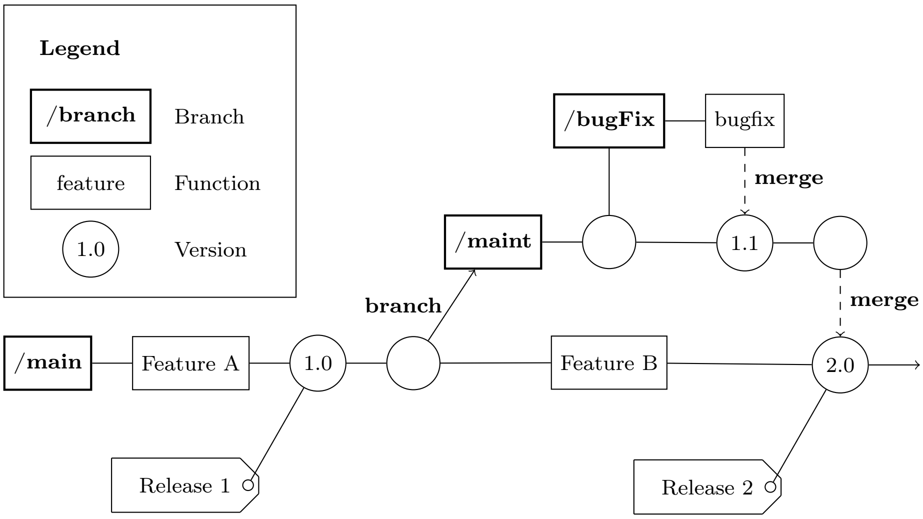

Figure 2: Exemplary codeline diagram representing one mainline development

Although the telematics and communication technology for SOTA updates is available, OEMs are still delaying the introduction of SOTA updates as a commercial feature, due to two main challenges. The first challenge is the security concern, which arises through opening the car to the outer environment with the possibility of reprogramming its embedded systems making it vulnerable for hacking attacks. The second one is due to the high number of variants and configurations within automotive product lines making it difficult to deliver safe and consistent updates for all existing combinations within a typical EE architecture. This question was addressed by the conducted survey.

2.3 Software Configuration Management

Software Configuration Management (SCM) is a general term used to describe the different activities required for the management of the software parts of a product along its whole life-cycle including the phases of development, production and maintenance during after-sales. It comprises configuration identification, configuration control, status accounting, review, build management, process management, and teamwork [BA02].

Considering the complexity of modern automotive systems, their increasing software functionalities and the significant size and spreading of the teams involved in their development, optimising SCM activities with respect to time, costs and quality is a continuous challenge for tier-one suppliers and OEMs.

Usually software changes of a module or system made between releases are documented and tracked inside a so called *codeline diagram* illustrating the main development branch with the evolution of system releases. An example of a simple codeline diagram is depicted in Figure 2.

In order to apply software changes to the system and consequently get new versions, which can be defined as new system releases as described in section 2.2.1, consistency checks of the relevant configurations must be undertaken using different possible methods. One common method is the use of dependency descriptions such as interfaces or contracts between composed configuration items at the same level of hierarchy [HFS04].

In addition to consistency checks, regression testing is an essential step before each system release to ensure that the system functions as expected and that the introduced changes did not lead to other still unrecognized system errors or failures. When a failure happens during regression tests, debugging and further unit tests are required to determine what component or interface broke [BA02].

Besides version and configuration management, OEMs in the automotive industry must deal with large variant spaces of their systems. The high number of system variants arises from different country regulations, multiple equipment models, and different construction and realisation alternatives [Pre13]. Especially the realisation alternatives with various ECUs, sensors and actuators are of main interest when dealing with EE architectures. Although several methods and tools like PREEvision[1] already exist to manage variants during the development, dealing with system variability with the shortening update cycles of software due to e.g. software bugs is nowadays an important challenge.

2.4 Related Work

Different further surveys and studies about the current and future trends within the development and maintenance of cyber-physical systems have been conducted. In the survey of Braun et al. [BAG15], interviews with experts from German suppliers, OEMs and research institutes were carried out to identify the current development trends of EE architectures in the automotive industry. Here, the need for new software architectures with service-oriented functions is seen as a consequence of the increasing safety and connectivity functions, and the update capability in the field is accepted as a necessary feature in the future to keep pace with the fast IT innovations evolution. Besides, the amount of system derivatives and increasing number of vehicle segments is presented as a challenging factor for the current development processes.

Another important example of complex cyber-physical systems with similar challenges as for vehicles is the domain of automated production systems as described in [Li+12]. In this work, the complexity of those interdisciplinary systems with mechanical, electrical/electronic and software parts together with the different frequencies of the involved innovation cycles are seen as the main challenge in the system and release development. An expert survey has been conducted to confirm this hypothesis, and the results showed that paradigm changes are the main innovation triggers to solve this complexity. However, missing solutions for module management as well as the diversity of variants and versions were identified as main innovation inhibitors [Li+12].

An empirical study by Berger et al. [Ber+13] based on a questionnaire analysed variability modelling in industrial software product lines. Different application domains such as automotive, industrial applications and aerospace have been considered to identify the different notations and tools used to model system variability. The scales and constraints of those common models were presented and discussed. Subsequently, major challenges such as dependency management and model evolution have been specified.

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

3 Conducted Survey

3.1 Research Method

We selected a quantitative research method to compare answers on challenges, the current state-of-the-art and expected future trends. We employed a questionnaire to collect the data. To count responses, we offered closed questions with a predefined answer set, but also provided open questions for additional concerns of the participants. Open questions were especially used for identifying future trends. As these 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 [Lik32] or derivations of it. Hence, we can sum answers across participants, compare selected answers and, thus, perform statistical tests on the results, but cannot compute means etc.

According to the classification by Shaughnessy et al. [SZ85], we performed a cross-sectional study, which gathered data at one specific point in time. We did not collect data over time to identify certain developments, but used specific question to identify an expected or predicted trend regarding specific topic. 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.

We performed a quasi-random selection as the sampling method for choosing the participants. We did not perform a selection based on certain properties of the persons, but were also not able to perform a completely random selection from all persons of interest. Due to their restricted reachability, we had to employ open mailing lists and personal contacts to find participants.

3.2 Design of the Survey

The design of the survey started with an identification of the information that we want to collect and a classification of that information into different categories (see Figure 3). We identified six categories of interest and developed 20 questions with certain sub-questions in total. All but two of these questions had a closed answer set. The developed questions were first reviewed by an additional expert on empirical methods outside the automotive domain. Here, we gave special attention to exclude effects through a specific ordering of the questions. The categories and questions were ordered in a way, such that they start from a survey on the state-of-the-art towards an identification of future trends. Then we used a pretest group of 3 pseudo-participants to check for understandability.

Participants were contacted in an open call over mailing lists and personal contacts into established German automotive companies. To ensure anonymity of the participants, evaluated data were not correlated with personal data.

1.	Participant's environment (field of work and product)	*(2 questions)*
2.	State-of-the-art update and variant management	*(3 questions)*
3.	Field updates and software over the air updates	*(5 questions)*
4.	Consistency of updates	*(3 questions)*
5.	Multi-domain development and design space	*(5 questions)*
6.	Conclusions regarding limitations and desires	*(2 questions)*

Figure 3: Categories of survey questions

3.3 Implementation

The questions have been formulated in German language because the target for the conducted survey was the German industry. Considering the long tradition and the established know-how of the German automotive industry, which is nowadays investing much in research to integrate the new automotive trends such as autonomous and connected driving into their products, we can assume without a loss of generality that the identified issues in our survey would be the same or very similar to the ones faced in other countries.

After finalising 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.

3.4 Content of the Survey

The main goal of the survey was to identify the current state-of-the-art and the challenges of the release and configuration management in the automotive industry. The focus is set on software/hardware 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 degrees of freedom to realise 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.

4 Results and Discussion

In this chapter, the results of the conducted survey are presented and discussed in order to identify the main current and future challenges when dealing with updates and variant-rich systems in the automotive industry.

4.1 Participant's Environment

51 participants from different segments took part in the online survey with most participants being engineers of OEMs and tier-one companies. Figure 4 shows the numbers of the participants per segment.

Most participants develop and maintain software (57 %) and hardware (45 %) components. In addition, an important part of the participants is responsible for EE systems (35 %) or EE 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.

[2] `https://www.limesurvey.org/`

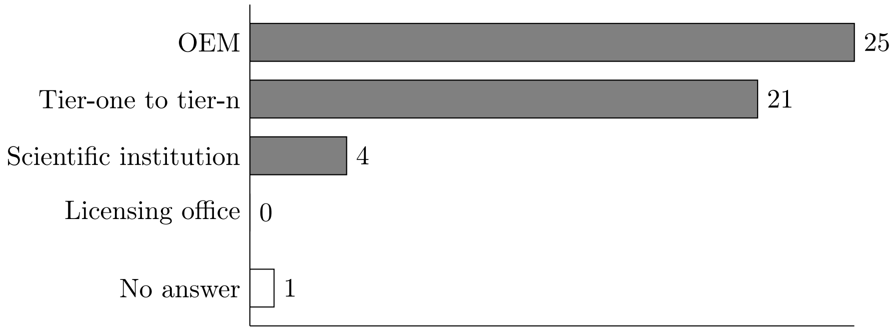

Figure 4: Segments of the survey participants

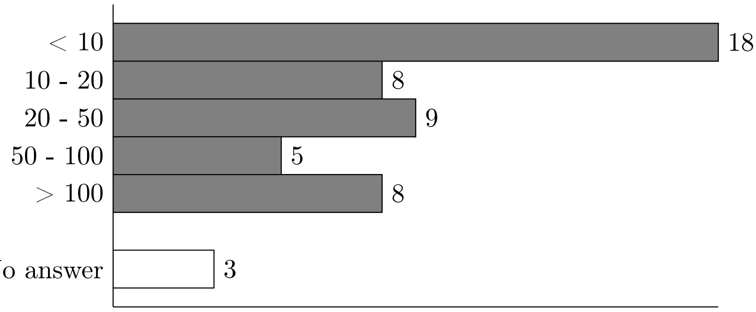

Figure 5: Number of system variants managed by the participants

4.2 Identified Current Challenges

4.2.1 Variant Management

The majority of the participants (35 %) are maintaining a small number of variants (less than 10), but 18 % are managing 20 to 50 and 16 % are even responsible for more than 100 system variants. To avoid misunderstanding in the case of managing more than one system, a hint to give an estimated mean value of the variants numbers for the different managed systems has been added to the question. Figure 5 represents the distribution of the variants numbers managed by the participants.

4.2.2 Dealing with Software Updates

New releases of automotive software during the development normally deal with corrections of bugs or system optimisations 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 EE 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.

The stated frequency of releases in form of new EE states for products in the field can be expected to depend on the functionality of the component to be updated. For example, an update of the navigation system as software will

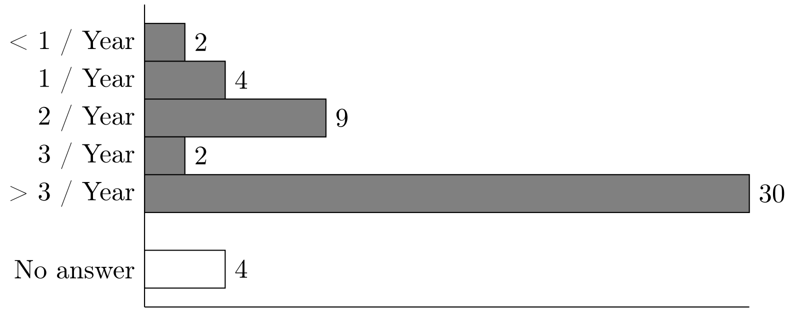

Figure 6: Frequency of system releases during development

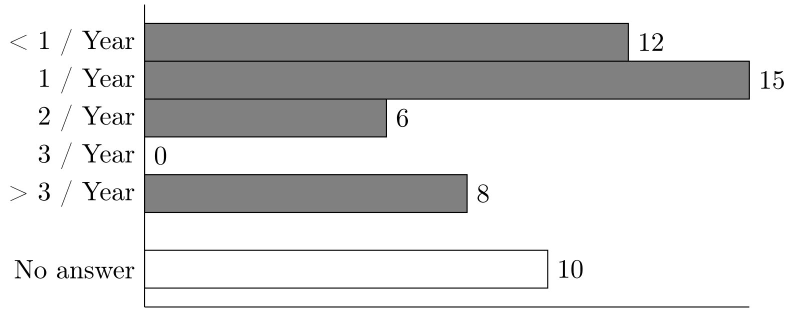

Figure 7: Frequency of system releases for products in the field

potentially be performed more frequently than an update of a more safety-critical component or even the hardware of a sensor. The responses of the participants regarding the release frequency for products in the field reaches from less than once a year to more than three times per year. 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 times a year, as depicted in Figure 7. This shows that most automotive systems are still getting no or seldom new releases. The updates with higher release frequency of three times per year can potentially arise from updates of not safety-critical components, as mentioned above. To confirm this hypothesis, one would have to analyse the correlation between the updated type of product and the release frequency.

The majority of the participants reported that the effort to validate an update is equal (29 %) or even smaller (33 %) than the effort needed to validate the initial product. Nevertheless, 28 % of them revealed that this effort is higher or much higher than within the initial validation process.

When asked if field updates with short life-cycles will gain more importance for seamless connected mobility, almost all participants (94 %) agreed on this statement. A reason for this could be the increasing degree of software realisations of vehicle functions and the resulting higher probability of software bugs.

35 % of the participants reported that the offered current time span for field updates is relatively short and ranges between one and three years. This is probably some kind of after sales guarantee for repairing all potential malfunctions or bugs directly by the OEM. Nevertheless, 20 % are offering filed updates

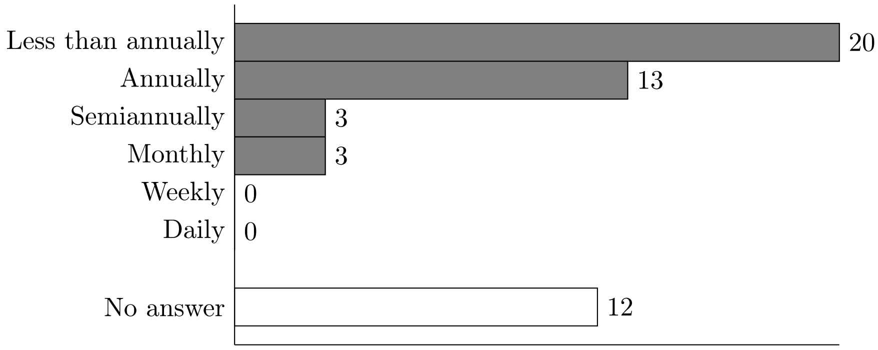

Figure 8: Frequency of updates in the field

for at least 7 years. This time span is expected to increase to the whole vehicle life-cycle when introducing SOTA updates as core system feature.

The majority of the participants revealed that the frequency of field updates is still very low and estimated at once a year or even more rarely as shown in Figure 8. These results are for the most part in concordance with those regarding after sales releases as depicted in Figure 7. In fact, 80 % of the participants who reported conducting field updates less than once a year revealed also 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. As a hypothesis for future studies, we formulate that the number of releases for products in the field is usually larger than the number of executed updates, which is already indicated by this comparison. This means that not every release will be updated. Some releases can be for example developed within research and development sections for the sake of improvement and optimisation of current products resulting in concept systems which can be integrated in the next product generations.

The main given reasons for current updates are bug fixes and function improvement. 88 % of the participants stated that bug fixes are one of the main reasons for software updates and 67 % stated this also for function improvements. Costs optimisation and component availability are, according to the questionnaire responses, less relevant reasons for an update than the ones mentioned above.

The participants were asked for the main reasons for faced inconsistencies of their products that occur in their development processes (see Figure 9). 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 %).

In order to safeguard the compatibility of updates with the existing EE environment, consistency checks are essential and crucially determining the update quality and safety. 45 % of the participants reported that these checks are representing at least 50 % of the work effort needed to remediate occurring inconsistencies.

Regarding SOTA updates, we presented three statements to which the par-

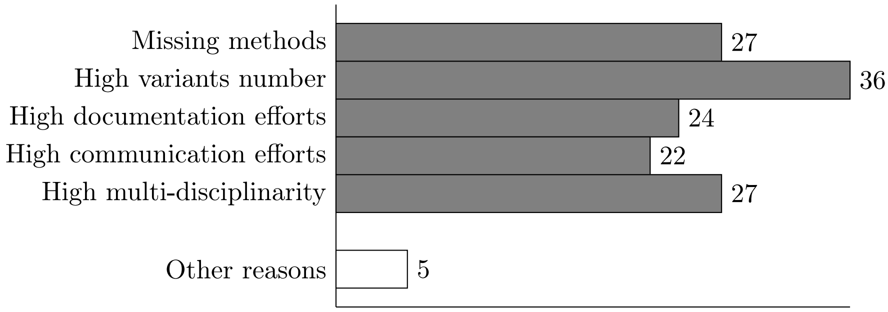

Figure 9: Reasons for inconsistencies during product development

Statement	fully agree	agree	rather agree	rather dis-agree	dis-agree	fully dis-agree
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

Table 1: Agreement with statements regarding the current state and future of SOTA updates in the automotive industry (in percent of all participants)

ticipants could indicate their agreement. The statements and the distributions of answers are shown in Table 1. Most of the participants (77 %) revealed that their institution is currently working on their introduction 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 will represent in this case an important part, 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.

4.2.3 Multi-Domain System Design

During the development of complex automotive systems, the question whether to realise 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 analysing the design space with hardware, software and mechanical models considering and evaluating different possibilities. When considering only hardware and software, this methodology is known

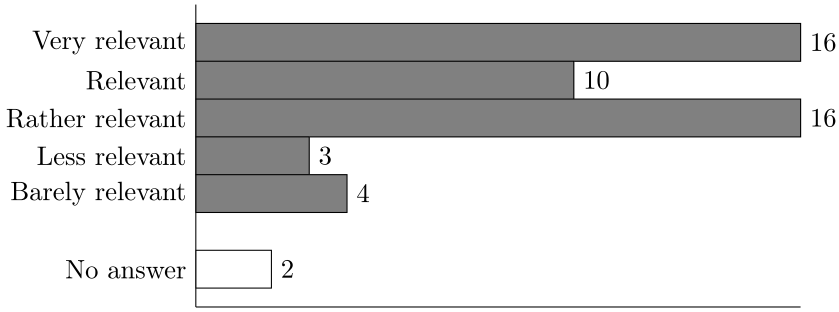

Figure 10: Importance of decision freedom for multi-domain function realisation within automotive institutions

Statement	agree	rather agree	rather dis-agree	dis-agree
More degrees of freedom for design decisions lead to higher quality of the product.	19.6	**43.1**	21.6	2.0
More degrees of freedom for design decisions lead to more efficient development of the product.	27.5	**43.1**	11.8	3.9
More degrees of freedom for design decisions lead to more efficient maintenance of the product.	19.6	**52.9**	9.8	3.9

Table 2: Agreement with statements regarding the expected influence of degrees of freedom for design decision for multi-domain function realisation on the product (in percent of all participants)

as hardware/software codesign, where the design of both hardware and software system parts is made in parallel. 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 optimised 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 10.

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 Table 2.

4.3 Future Challenges

The more software influences the value of a car and the more services and functions are based on software, the more error prone the complete system of a vehicle gets. In addition, the developers work in separate groups all over the world, which is a challenge for the synchronisation of a common release stand.

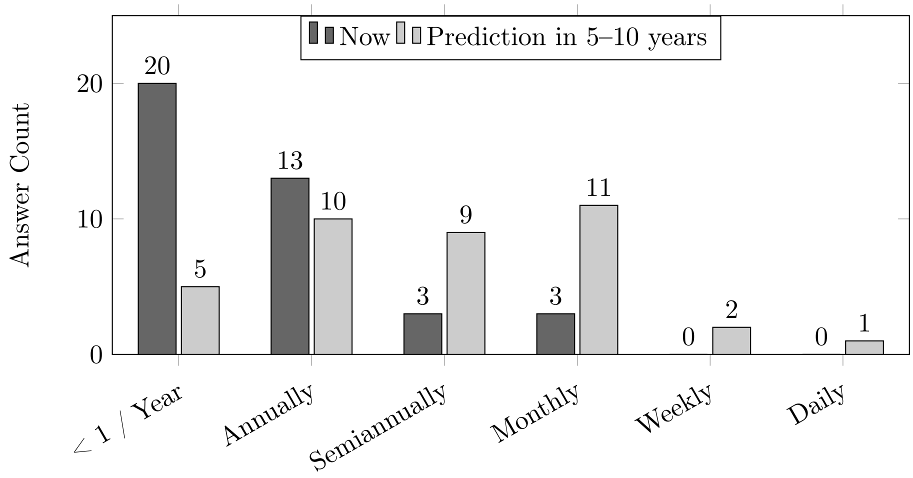

Figure 11: Predicted evolution of the update frequencies in the next 5 to 10 years

This, again, is an error-prone process. The consequence will be bug fixes during use. Furthermore, the customer's wishes for innovations and individualisation demand for software updates in shorter and longer terms. This results in a need for more frequent updates that flashing a software at the end of the assembly line with a 6 months refinement cycle. These facts have been confirmed by the conducted survey, which indicated, according to the participants, that the frequency of field updates will significantly increase in the upcoming five to ten years, as illustrated in Figure 11.

Besides, the reasons for the updates will partially change. While bug fixes and function improvement will continue to increase, the costs optimisation as a reason for updates will play a greater role, as reported by the survey participants. 67 % of the participants agreed that bug fix as 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 optimisation's updates, their growth is expected by approximately half of the participants, and only 8 % think that their importance will decrease.

We expect that, in the end, we will have updates in the field by means of over the air communication because updating frequently in a workshop is much too expensive.

4.4 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 variety exceeds very rapidely the 10 million barrier [SSS07]. 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 wished more process, methods and tools support for the variants management, which is considered in the survey as the main challenge in the release and configuration management of modern and future automotive systems.

Effective and efficient checks before software flashing are one of the only chances to solve the problem of consistency. The risk that flashed software influences the existing one and causes failures or even severe situations if consistency is not ensured is much too high. 61 % of the participants would like to be supported by more processes, methods and tools for performing consistency checks.

4.5 Threats to Validity

As in any empirical study, there are threats to its validity. 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 generalisability of the results.

4.5.1 Internal Validity

The main threat here is that different participants had different understandings of the questions or the potential answers. This would result in answers that cannot be summarised into one value. 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 the potential answers given also helps to understand the meaning of the questions as intended by the authors. The unique and identical understanding of the potential answers 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 internal threat to validity could be the ordering of questions. Here, the danger is that answering one question primes certain terms or concepts in the mind of the participant, which influence the answers to the next question. 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. To the best of our knowledge and imagination, we see no effects of the chosen ordering of the questions, i.e., we do not see that the presence of a question affects answers to later ones.

Further, participants may be biased to certain answers. On the one hand, 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. However, through the given anonymity, we can assume that this bias is limited, as the participant cannot expect any positive or negative consequences.

4.5.2 External Validity

Here, 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 generalisability 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 finding 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 manufacturers of low quality cars may differ, as well as potentially for cars with an electrical power train. This is not necessarily a matter of the power train technology, but also with the maturity of the industry. 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 organisation 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 will only be applicable in regions, where car industries also use such design properties and approaches, which is usually the case for more mature car manufacturers.

Another threat to the external validity of our survey 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 (mailing lists, personal contacts) and also the contacts of different persons for distribution, so that we 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 limits regarding the generalisability of our results to arbitrary car manufacturing regions, this could also reduce the generalisability for the Southern German industry, of which we selected the participants. There are two reasons why we do not consider this as a thread 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.

5 Conclusions and Future Work

The growing importance of software in the realisation of new mobility applications such as automated driving demands for fundamental changes in software development for modern vehicles. The number of bugs not recognised before usage will rise. In addition, the customer's wish to enhance existing functionality is triggered by communication means, such as handhelds. Therefore, it will not suffice to flash software into a car only once at the end of the assembly line anymore. Software over the air updates are an upcoming demand. But due to vehicle variants of one type and software releases in production every six months, the variety of software combinations in the field is strongly growing.

We conducted a survey in the German automotive industry to identify the state-of-the-art in update and variant management for vehicles and the current challenges regarding the introduction of software over the air updates and the assurance of consistency of these updates with the rest of the vehicle system. All participants demanded for new ways in the validation of software before any updates are sent into the field. Testing the consistency with the large number of existing software releases is the biggest challenge. If this consistency cannot be guaranteed and the interaction of existing software with an updated one will cause errors, in the best case cars will not be usable anymore. In the worst case, especially in the application field of automated driving and vehicle dynamics, this can cause severe damage.

To ensure this demand of consistency, new approaches for the development of vehicle systems have to be developed. In our survey, we identified that methods such as hardware-in-the-loop are already established for testing updates before their deployment. Nevertheless, most participants agree with a need for methods going further and supporting the development process before the validation step. The majority of the participants sees a demand for exploring domain-spanning design spaces, e.g. considering both hardware and software simultaneously, to develop appropriate solutions for specific problems.

Based on the results of our survey and its evaluation presented in this report, in future work the collected data should be further analysed regarding correlations between certain metrics. An example for such a potential correlation is the dependency between the numbers of managed variants and the effort for ensuring and validating consistency. These were both covered by individual questions and could thus be analysed for correlation.

Furthermore, a catalogue of hypotheses regarding the development of software updates and variant management in the automotive industry should be defined and tested against the data collected in this survey. Those hypotheses should especially go beyond those directly arising from the questions in the survey, as they were already discussed in this report.

Some correlations could only be speculated about in this report because required data was not collected in the conducted survey. One example for this is the dependency between the update frequencies and the type of a product and especially how safety-critical it is. Together with a prognosis for the development of the relation between safety-critical and non-safety-critical components in cars, this would yield a prognosis for the development of the update frequencies of products in the field.

Acknowledgements

We would like to thank all participants for their willingness to participate and their time spent in filling out the survey. We thank Anne Koziolek and Eva Reussner for their helpful consultancy on empirical research methods. We also want to thank Nikola Bursac and Max Kramer for reviewing the first version of the survey and for their precious improvement's suggestions.

References

[Ant+13] V. Antinyan, M. Staron, W. Meding, P. Österströ, H. Bergenwall, J. Wranker, J. Hansson, and A. Henriksson. "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, pp. 1–15.

[BA02] S. P. Berczuk and B. Appleton. *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*. Addison-Wesley Longman Publishing Co., Inc., 2002.

[BAG15] L. Braun, M. Armbruster, and F. Gauterin. "Trends in Vehicle Electric System Design: State-of-the Art Summary". In: *2015 IEEE Vehicle Power and Propulsion Conference (VPPC)*. 2015, pp. 1–6.

[Ber+13] T. Berger, R. Rublack, D. Nair, J. M. Atlee, M. Becker, K. Czarnecki, and A. Wąsowski. "A Survey of Variability Modeling in Industrial Practice". In: *Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems*. VaMoS '13. ACM, 2013, 7:1–7:8.

[BJ15] C. Bird and E. Juliussen. *Improving software, reliability & innovation*. Tech. rep. IHS Technology, 2015.

[Bro06] M. Broy. "Challenges in Automotive Software Engineering". In: *Proceedings of the 28th International Conference on Software Engineering*. ICSE '06. 2006.

[HFS04] C. Heinisch, V. Feil, and M. Simons Daimlerchrysler. "Efficient configuration management of automotive software". In: (2004).

[Li+12] F. Li, G. Bayrak, K. Kernschmidt, and B. Vogel-Heuser. "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 (INCOM)*. 2012, pp. 1077–1082.

[Lik32] R. Likert. "A technique for the measurement of attitudes." In: *Archives of Psychology* 140 (1932), pp. 1–55.

[Mul+04] K. D. Muller-Glaser, G. Frick, E. Sax, and M. Kuhl. "Multiparadigm Modeling in Embedded Systems Design". In: *IEEE Transactions on Control Systems Technology* 12.2 (2004), pp. 279–292.

[OG14] H. A. Odat and S. Ganesan. "Firmware over the air for automotive, Fotamotive". In: *IEEE International Conference on Electro/Information Technology*. 2014, pp. 130–139.

[Pre13] P. Prehl. *Systematisches Variantenmanagement für Elektrik/Elektronik-Architekturen im Kraftfahrzeug.* Shaker, 2013.

[Sax08] E. Sax. *Automatisiertes Testen Eingebetteter Systeme in der Automobilindustrie.* Ed. by E. Sax. Carl Hanser Verlag GmbH & Co. KG, 2008.

[SSS07] H. Schleich, J. Schaffer, and L. F. Scavarda. "Managing complexity in automotive production". In: *19th international conference on production research.* Vol. 100. Citeseer. 2007.

[SZ85] J. J. Shaughnessy and E. B. Zechmeister. *Research methods in psychology.* Alfred A. Knopf, 1985.