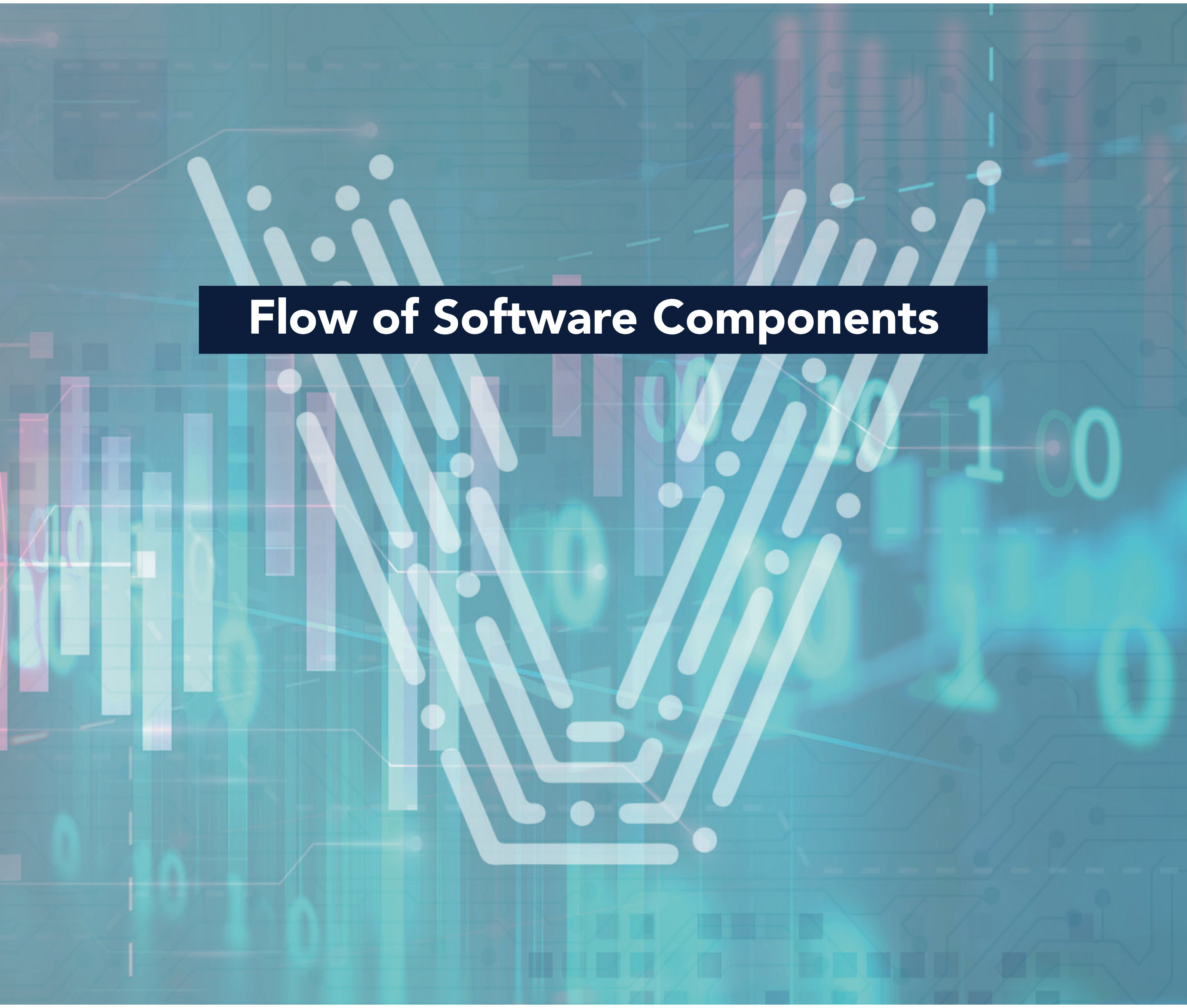
prostep IVIP

White Paper



Flow of Software Components

prostep ivip White Paper

Results of the "Flow of Software Components" survey

Collaboration processes in software development
for the exchange of software components between companies

This White Paper was created within the framework of the
prostep ivip and VDA project group "Standardization Strategy Board"

Disclaimer

prostep ivip documents (PSI documents) are available for anyone to use. Anyone using these documents is responsible for ensuring that they are used correctly.

This PSI documentation gives due consideration to the prevailing state-of-the-art at the time of publication. Anyone using PSI documentations must assume responsibility for his or her actions and acts at their own risk. The prostep ivip Association and the parties involved in drawing up the PSI documentation assume no liability whatsoever.

We request that anyone encountering an error or the possibility of an incorrect interpretation when using the PSI documentations contact the prostep ivip Association (psi-issues@prostep.org) so that any errors can be rectified.

Copyright

Table of Content

Figures

1 Introduction

The requirements relating to how software components are to be handled in collaborative development scenarios are currently often defined individually by companies in the automotive industry. There are no standardized cross-enterprise processes for the development and deployment of software components at carmakers and companies in the supplier industry. Against the backdrop of increasing technical and legal complexity in the development of safety-related software, the prostep ivip Association has conducted a survey to determine how carmakers and their tier 1/tier 2 suppliers currently handle the development and management of software components and what the current state of the art looks like at the participating companies. The survey, entitled Flow of Software Components (FoSC), was organized and conducted by the Standardization Strategy Board (SSB). The Standardization Strategy Board is a joint project group set up by prostep ivip and the VDA to address issues relating to the need for standardization in the context of cross-enterprise collaboration in product development and manufacturing.

The survey and its results will be used to develop a common understanding of the constraints and shared expectations regarding technical and legal issues between OEMs and tier 1/tier 2 suppliers. It is also intended that the survey results provide a basis for providing assistance for the future design of cross-enterprise software development in projects.

The survey prepared and conducted by the Standardization Strategy Board comprised a total of 26 questions. Some questions were closed-ended with a predefined list of response options, while others provided not only specific predefined response options but also the option of a non-specific response ("Other"), which allowed participants to respond in their own words. The original responses are not provided in their entirety here; a summary of the content is provided instead. Some questions allowed the selection of multiple response options. In this case, the sum of all the responses may be greater than the number of survey participants. Questions that allowed multiple responses are indicated in the caption by ("multiple choice"). As some survey participants may not have answered all of the questions using the predefined response options, the number of responses may be smaller than the number of survey participants. The total number of responses to a question is indicated on the right-hand side of the bar charts/diagrams. Questions with predefined responses are displayed as ungrouped, unstacked and non-standardized horizontal bar charts.

The question of OEMs' expectations regarding supplier response times in the event of software errors with different severity levels from the OEMs' perspective and the same question from the suppliers' perspective represents a special case. The results of these two questions are displayed together in a matrix and each individual response is represented by a blue or green dot (see Figure 8).

The survey involved the participation of 28 people from 19 companies, which can be divided into two groups. 17 respondents (61%) stated that they were employed by a supplier, while 11 respondents (39%) stated that they worked for an OEM. Figure 1 shows this distribution in a bar chart.

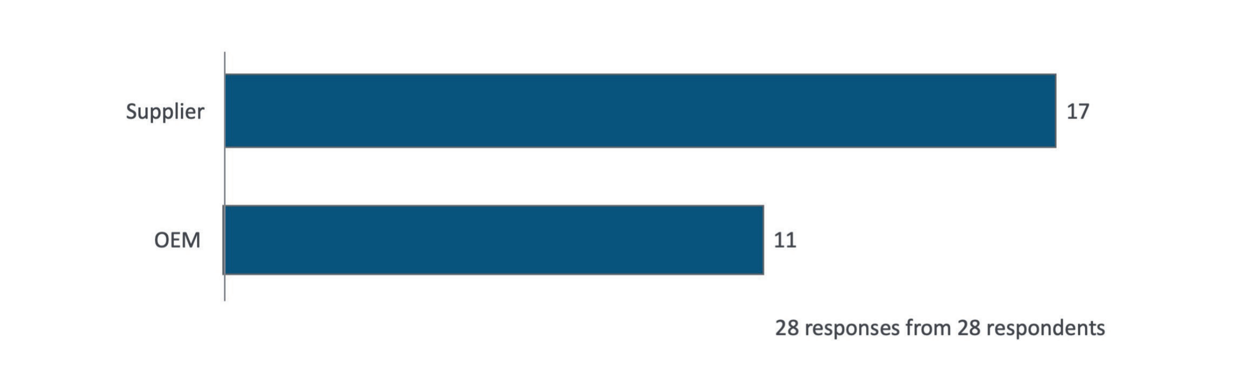


Figure 1: Is your employer a supplier or an OEM?

This white paper first of all examines the importance of software as a component for automotive products and the resulting challenges this poses in the context of software development and software component management in the automotive industry. The survey and its key results are then presented.

2 Challenges in automotive software engineering and software management

Over the course of the last 30 years, not only mechanical and electrical/electronic components have found their way into cars but also software components. While the mechanical and to some extent also the electrical/electronic components are still primarily tasked with giving the vehicle its structure and putting torque on the road, driving and comfort functions are increasingly being determined and implemented by the use of software. This development began in the 1980s and 1990s with comparatively simple driver assistance systems, such as the anti-lock braking system (ABS) and the electronic stability program (ESP), and has since evolved to include the software-controlled coordination of the behavior of engines, transmissions, chassis and other systems. And then there are the numerous systems that serve to enhance comfort and infotainment. Examples here include air conditioning, sound systems and navigation systems.

For the past ten years or so, development in the direction of autonomous driving has been picking up speed. When it comes to autonomous driving, there is more involved than merely assistance systems that provide the driver with support. In fact, software systems assume complete control of almost all driving functions. This poses not only an enormous challenge in terms of the safety of automotive software components but also in terms of the speed at which software-based information processing is performed. Software systems for implementing and supporting autonomous driving must also be highly reliable and – an aspect that should not be underestimated – gain the trust and acceptance of customers.

Today, software is not only used to support automotive functions within vehicles. It is also increasingly being used to support communication between the automobile and systems and infrastructures outside the vehicle such as, for example, communication with parking and traffic guidance systems and navigation satellites, as well as communication with mobile devices or the backbone systems of carmakers and car repair garages. In some cases, these external systems and infrastructures are no longer under the direct control of the carmaker. Software systems connected to the outside world may also be vulnerable to outside attacks.

The variety of ways in which software can be used in cars, and thus also the wide range of options in automotive software development, is confirmed by the responses given by the survey participants, as can be seen in Figure 2.

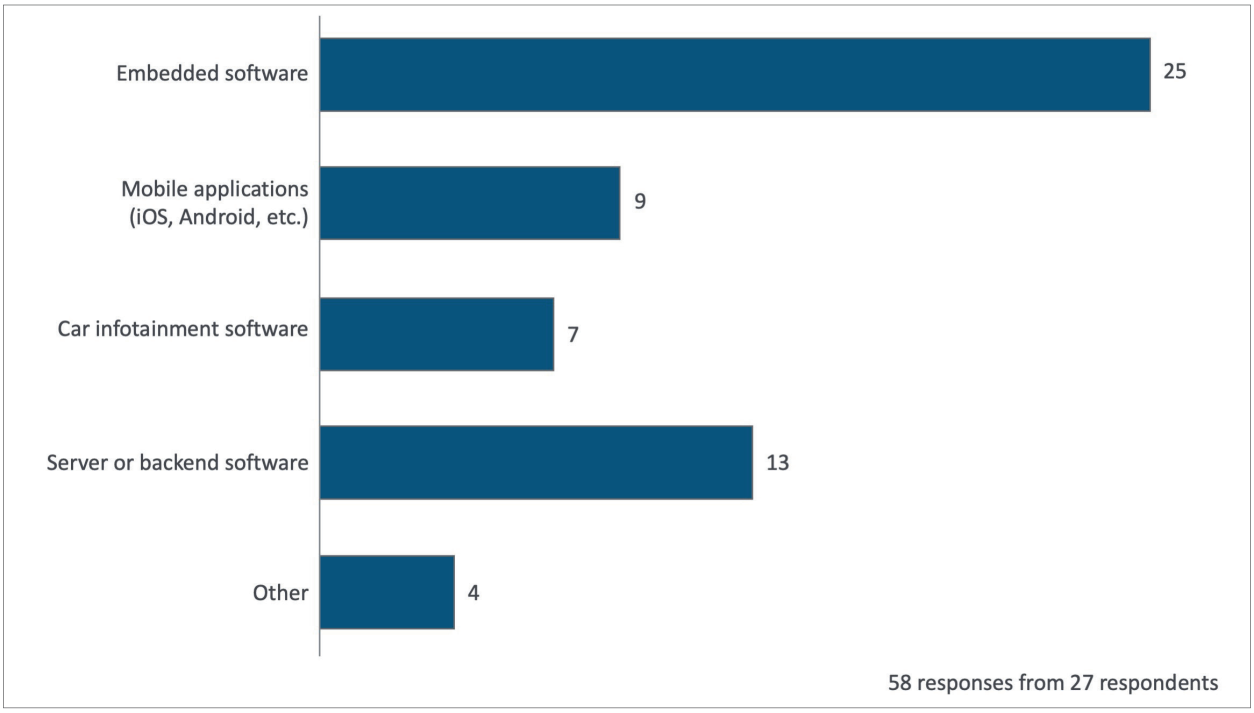


Figure 2: What types of product-related software are relevant to you (multiple choice)?

As things stand today, it is first and foremost embedded software components that are subject to strict safety requirements, both in terms of their functional reliability and protection against outside attacks (hacking). As can be seen in Figure 3, this is also confirmed by the participants in the survey.

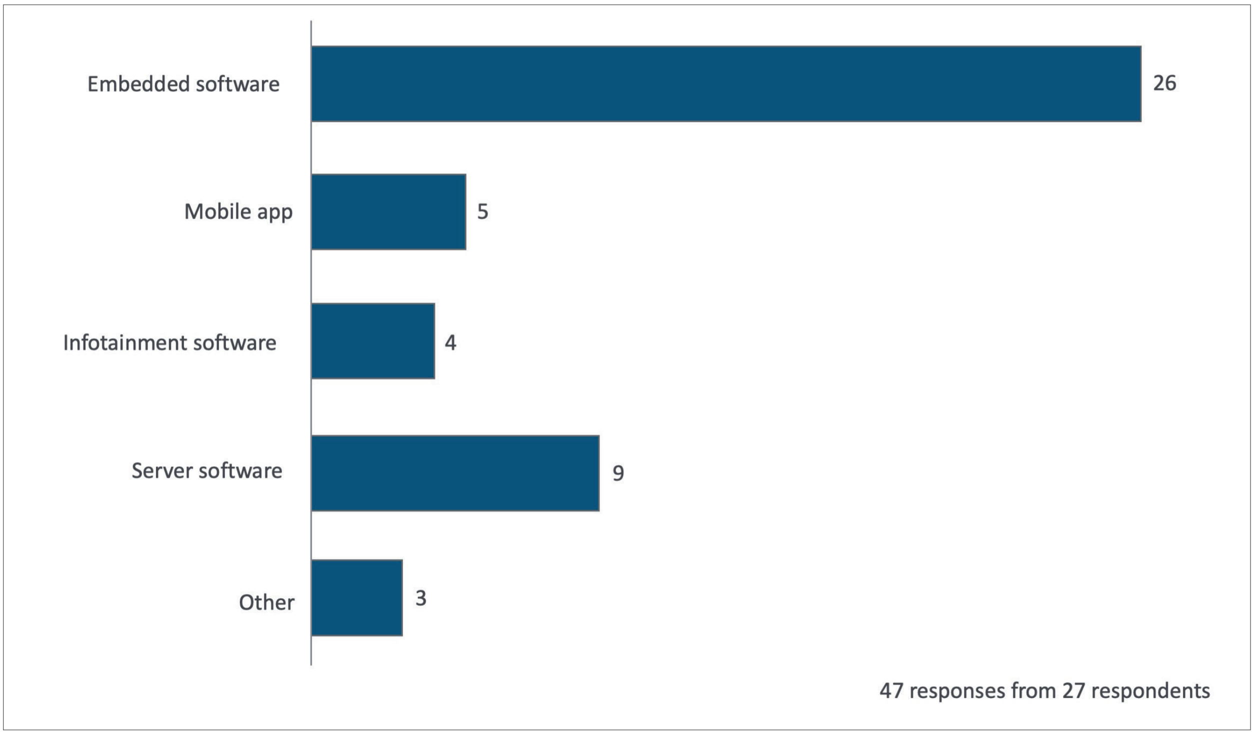


Figure 3: Which software is safety critical (multiple choice)?

The development of safety-critical software components deserves special attention in that faulty safety-critical software components can usually also impair safety-critical vehicle functions. The failure of such functions can, in a worst-case scenario, directly or indirectly pose a risk to life and limb of the vehicle occupants and of persons outside the vehicle, for example pedestrians and other road users in the vicinity of the vehicle. Critical software errors can be the result of both inadequately validated software development processes/software components and vulnerabilities in purchased or otherwise externally sourced software components. Another aspect is the interaction between different components in the vehicle as overall system. The aim here is to estimate the criticality from the perspective of the overall system. This poses a number of challenges for the manufacturers of software-controlled products. On the one hand, assessing the criticality of software components and taking appropriate measures to validate critical software components is vital to quality assurance. On the other hand, product manufacturers must also deal with vulnerabilities identified before and after the products are sold and find an appropriate response for reducing or eliminating these vulnerabilities (mitigation). When assessing criticality, it is extremely important that all the companies and developers involved in developing the software components and the respective driving functions come to the same conclusion. This ensures that the respective development objectives can be achieved and that expectations regarding product quality, product validation processes and product development processes can be met. When dealing with identified vulnerabilities and potential risks, it is also important how and how quickly they respond once the vulnerabilities and potential risks have been identified or become known.

With regard to procuring software components from external sources, a distinction has to be made as to whether a commercial software component or a so-called FOSS component is involved. FOSS stands for free and open-source software. Particularly when using FOSS, compliance with the respective license terms, which can be extremely varied and complex, is important. Since FOSS software components can normally be obtained and used by all interested

parties (although they are not necessarily permitted to do so), companies are faced with the challenge of having to manage and regulate the use of FOSS, on the one hand, and ensuring compliance with the respective license terms when using the FOSS, on the other.
Software components are just as much a part of the product as all the other components and must ultimately be handled in the same way as all the other components in the product lifecycle management (PLM) processes. Software components are however quite volatile. This means that they may be subject to much faster change than mechanical components, for example, and may also exhibit a higher level of variance. In addition, the functions in software components also change significantly once they have been delivered to the customer, for example as the result of over-the-air updates. The challenge that arises here is the assignment of the software components to the product structure of the product (bill of materials (BoM)). This is important because the unimpaired functionality and the safety of a driving function can only be guaranteed if the software is compatible with the mechanical, electrical and electronic components. The question of which version of which software in which configuration is also extremely important to the staff in the car repair garages. This applies not only to each vehicle model but under some circumstances also to each individual vehicle driven into a car repair garage.

Software development in the automotive industry, like most other development activities, is performed across different companies and is therefore associated with an extensive exchange of software components and information throughout the development process. And as in all other development disciplines, it is very important to set up the data exchange process in such a way that the right information can be exchanged in the right way at the right time. This is where standardization and harmonization typically come into play. Software development is also increasingly being performed in cross-enterprise cloud-based environments.

3 The survey and its results

The results of the survey are presented, explained and interpreted below, grouped according to topic.

- Use of free and open-source software
- Handling security vulnerabilities and potential risks
- Update strategies and software maintenance
- Relationship between software components and product structure
- Cloud computing and collaboration

The conclusions drawn from the survey results are bundled together for each topic.

3.1 Use of free and open-source software

The participants in the survey responded to the following questions regarding the use of free and open-source software (FOSS):

- How do you manage the use of Free and Open Source Software (FOSS) in software development? (Figure 4)
- How do you ensure compliance with license conditions? (Figure 5)
- How do you inform customers about the use of FOSS licenses? (Figure 6)
- How do you notice vulnerabilities of commercially purchased software or FOSS? (Figure 7)
- How do you deal with known vulnerabilities and attack points of commercially purchased software or FOSS?

The purpose of these questions was to determine the current status of processes and technology when it comes to dealing with FOSS and avoiding risks that could arise when using FOSS. However, focus in this context is not only placed on the risks posed by security vulnerabilities. When using free and open-source software, it is also important to comply with the licensing conditions, which may be relevant in a legal context.

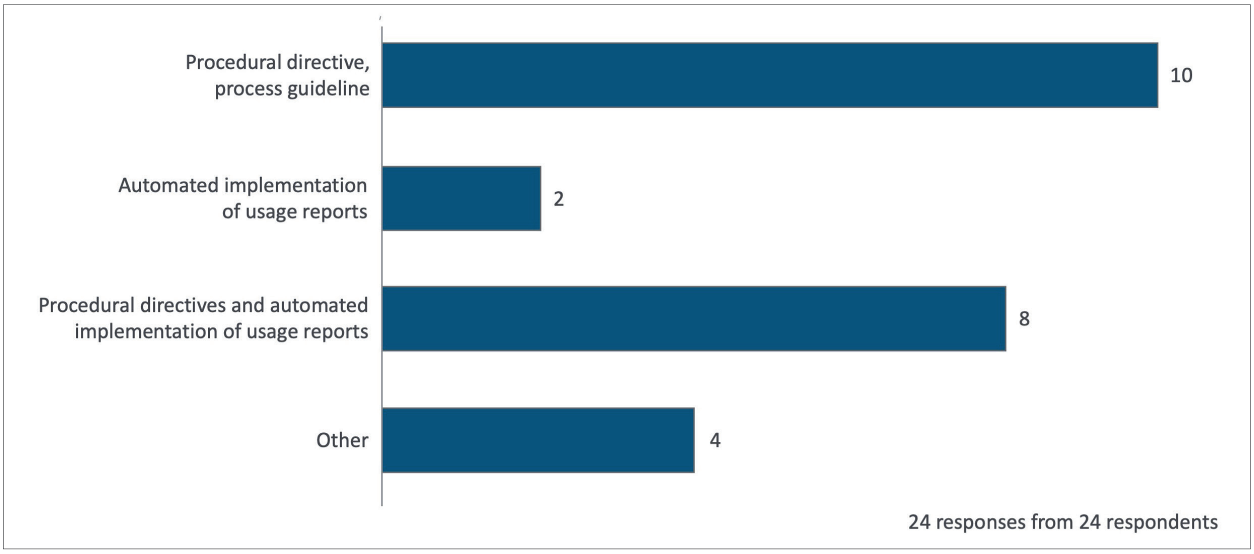


Figure 4: How do you manage the use of Free and Open Source Software (FOSS) in software development?

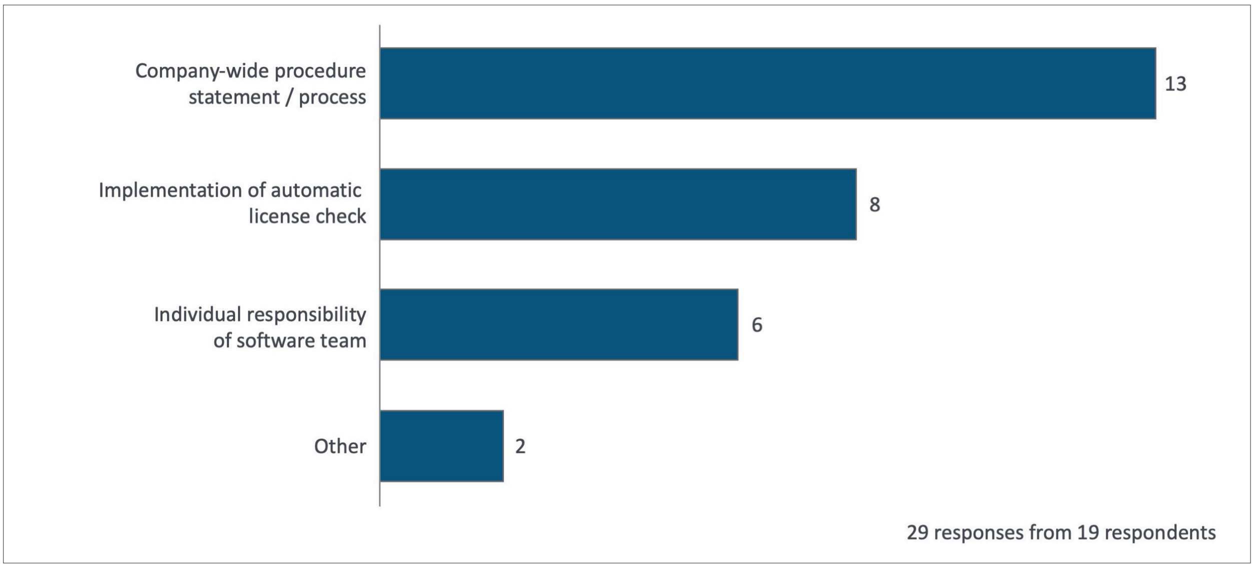


Figure 5: How do you ensure compliance with license conditions (multiple choice)?

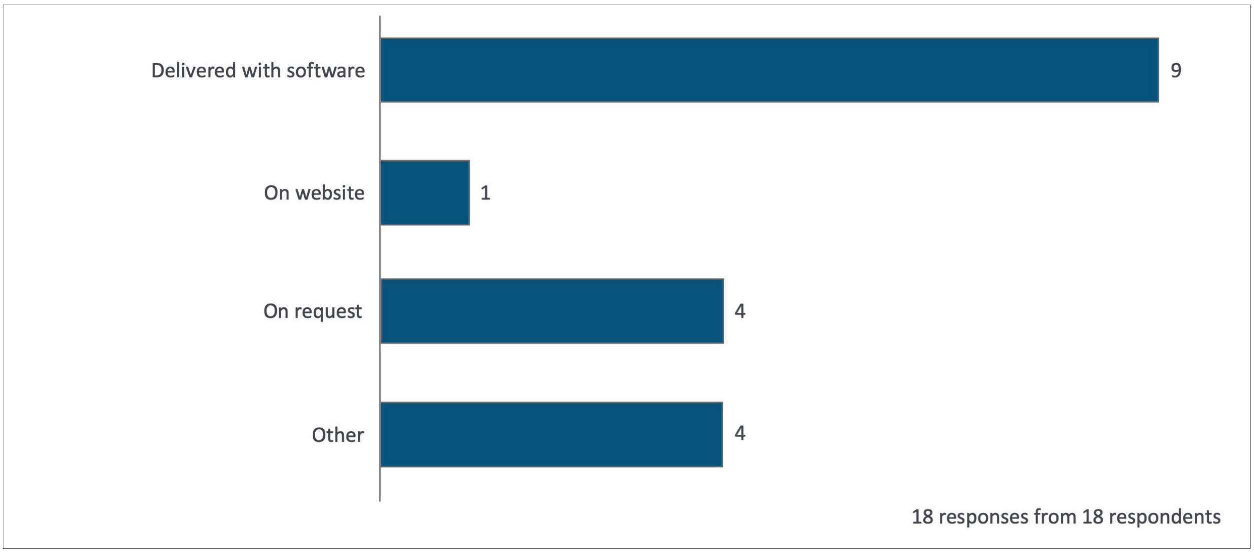


Figure 6: How do you inform customers about the use of FOSS licenses?

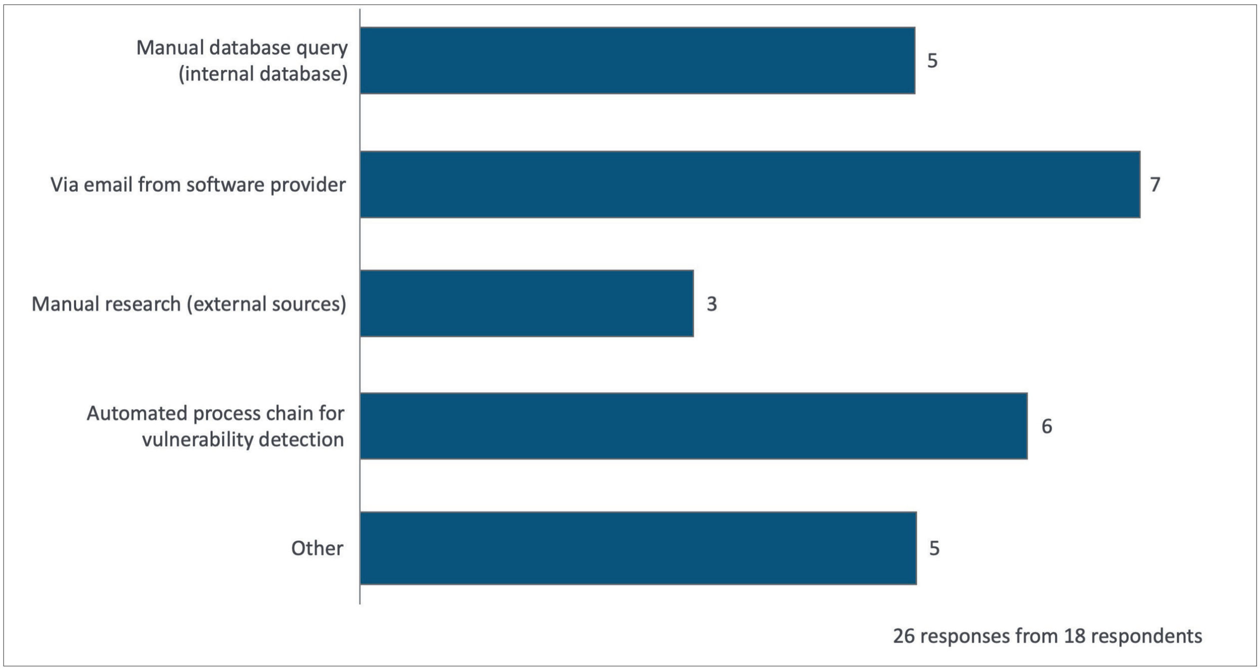


Figure 7: How do you notice vulnerabilities of commercially purchased software or FOSS (multiple choice)?

Figure 4 indicates that the use of free and open-source software (FOSS) in most companies is regulated exclusively by means of procedural directives. This was the response given by just under half of the respondents. About one-third of respondents indicated that compliance with procedural directives is also monitored using automated processes. This means that a total of approximately 75% of all respondents indicated that their companies have procedural directives regarding the use of FOSS. This means that only a few companies rely solely on monitoring via automated processes.

When FOSS licenses are used, many companies also regulate compliance with the license conditions by means of procedural directives. Significantly fewer respondents indicated that compliance with licensing regimes is the responsibility of the individual employees. However in these cases, the results do not indicate whether the responsibility lies exclusively with the persons taking action or whether additional procedural directives also exist. Just under half of respondents indicated that automated processes are used to monitor compliance with the license conditions. Figure 5 shows the distribution of the responses to this question.

When software products or products that include software components are sold to customers, the customers are usually also informed of whether and which FOSS licenses are used in the products. As Figure 6 indicates, a proactive approach to providing this information together with delivery of the software is predominant. Only a few companies use the option of hosting license information centrally on a website. Delivering the license information together with the software appears to be the conventional means of communication, while hosting the license information on a website would appear to be a bit more flexible as it allows the documentation to be updated on the website as necessary. In some cases, the response indicated that license information is provided upon request. In this case, customers must make an active effort to obtain the license information from the software vendors or product manufacturers as required.

When using FOSS and commercially purchased software, the handling of security vulnerabilities is a particularly sensitive issue. What is critical in this context is that security vulnerabilities are recognized in good time and information about any security vulnerabilities identified is provided quickly. The survey results shown in Figure 7 indicate that a number of different strategies are used to identify security vulnerabilities. Distribution among the predefined response options was relatively even. Active use is made of both internal data sources and external sources. Active communication between software suppliers and their customers via e-mail is also relatively widespread. In this case, the software suppliers' customers rely on the active involvement of the software suppliers. As this was a multiple choice question, the survey results do not clearly indicate whether this is the only communication channel or whether it is supplemented by one or more of the other measures. A relatively large number of respondents indicated that they use processes that were not included in the predefined responses ("Other").

When asked how companies deal with known vulnerabilities and points of attack in commercially purchased software or FOSS, it emerges that different strategies and measures, both organizational and technical, can be used. Many companies have teams or organizations that specialize in dealing with security vulnerabilities, or organizational processes for dealing with security vulnerabilities and risks exist. In some cases, automated processes for identifying risks and security vulnerabilities also exist. Some responses indicate that the respondents not only focus on the risks posed by embedded software and software products but also on the risks posed by the IT systems used internally. They mentioned the use of firewalls, for example.

3.2 Handling security vulnerabilities and potential risks

The survey participants responded to the following questions on how the security vulnerabilities and potential risks posed by software created in-house or externally sourced and deployed are handled:

- Which classification, if any, do you use to identify safety-critical software? (e.g. ASIL)
- If you are employed at an OEM, please name your expectations for fault response times according to the named severity levels (Figure 8)
- If you are employed at a Tier 1 supplier, please name what fault response times are expected by the OEM according to the named severity levels (Figure 8)
- What other severity levels do you use in your company?

The question about the classification schemes used to classify the safety levels of software components indicated that the ASIL scheme is clearly predominant. However, the results also indicated that the DO-178 standard is used in the aviation industry.

ASIL stands for Automotive Safety Integrity Level and is a classification scheme defined by the ISO 26262[1] standard. The components of a product are the subject of a hazard analysis and risk assessment performed using the methodology described in ISO 26262 (Functional Safety for Road Vehicles). The result of this assessment is the ASIL classification. Components, including software components, can therefore be classified according to four ASIL hazard levels, A to D, where ASIL A is the lowest hazard level and ASIL D is the highest. If the hazard analysis and risk assessment indicates that a software component requires an ASIL classification A – D, certain measures must be taken to minimize or eliminate the risks in accordance with the ISO standard.

DO-178, also known as Software Considerations in Airborne Systems and Equipment Certification, is a software development standard in the safety-critical aviation[2] industry. Compliance with this standard is required by U.S. and European aviation authorities. Five Development Assurance Levels (DAL), A to E, are used, where A stands for "catastrophic" and E for "no effect".

The standards ISO 25119 (Functional Safety for Tractors and Machinery for Agriculture and Forestry – Safety-related parts of control systems – Functional Safety)[3] and A-SPICE (Automotive SPICE) were also mentioned. A-SPICE is a domain-specific variant of the international standard ISO/IEC 15504 (SPICE). The purpose of Automotive SPICE is to evaluate the performance of development processes in the automotive industry[4].

The second and third questions in particular were designed to determine whether there is convergence between assumptions and expectations when it comes to response times in the context of software errors or whether assumptions and expectations diverge. The two survey results were combined in a single diagram to make it easier to compare the responses given by OEMs and those given by suppliers. The frequency of a given response is represented by blue dots for OEMs and green dots for suppliers.

[1] ISO 26262
[2] DO-178C
[3] ISO 25199 / DIN EN ISO 25119-1:2018-11
[4] Automotive SPICE

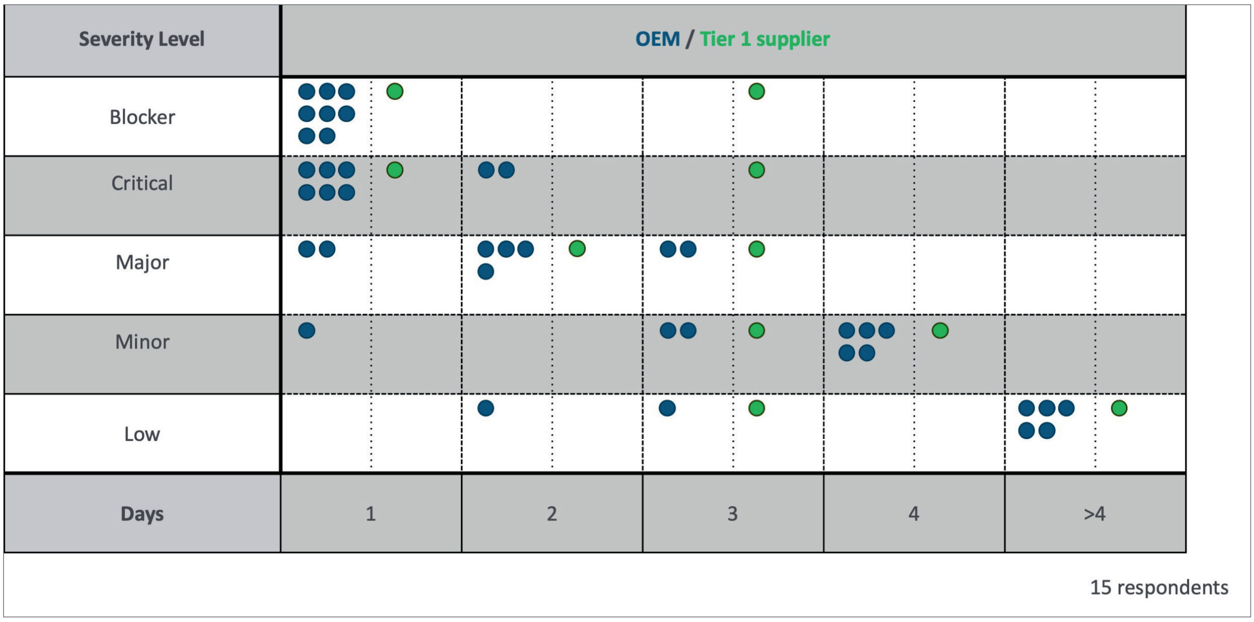


Figure 8: As an OEM/Tier 1 supplier, please name what fault response times are expected by each other according to the named severity levels:

When it comes to cross-enterprise collaboration, a consistent and coordinated picture of safety levels must exist to ensure that safety levels for the delivered or externally developed software and for the controlled vehicle function, for example, are compatible and that no required or regulatory measures are missed or omitted.

A shared expectation with regard to the response times for software errors in safety-critical software components is also important for guaranteeing the functional safety of products that have already been created and delivered to the end customer. The diagram in Figure 8 shows that there were significantly more responses to this question from the employees at the OEM surveyed than from the employees at suppliers. In Figure 8, each blue dot indicates a response from an OEM and each green dot indicates a response from a supplier. It is worth noting that the majority of participants come from the supplier industry. It can be assumed that the participants from the supplier industry in particular were unable to answer this question with any degree of certainty and therefore did not respond. There could be two reasons for this. On the one hand, many of the employees surveyed at companies in the supplier industry may not have any knowledge of OEM expectations with regard to response times due to the role they play in the company. On the other hand, it is also possible that there is insufficient documentation and communication of requirements and expectations with regard to response times, which means that there is potential for improvement here. The survey results do not, however, allow these assumptions to be substantiated.

Nonetheless, it can also be seen that the suppliers' awareness of the OEM's expectations is pretty much in line with the OEM's actual expectations. According to the OEMs, the more critical a severity level is (e.g. "Blocker", "Critical "or "Major"), the shorter the response time should be (e.g. one day for "Blocker" and a maximum of three days for the severity level "Major"). It is nevertheless worth noting that there was one response in each case, in particular for the severity levels "Blocker" and "Critical", which diverged from the OEM's expectations by one to two days. ("Blocker": plus two days and "Critical": plus one to two days). The lower the severity level, the longer the response time might be (e.g. four days for the severity level "Minor"). At the lowest severity level ("Low"), the response time could be longer than four days.

3.3 Update strategies and software maintenance

The growing proportion of software components in vehicles and products means that the maintenance of software components is becoming increasingly important. Maintenance measures must include updating the software. The following questions answered by the survey participants focus on this set of issues.

- How is the update of software done? (Figure 9)
- Who is responsible for the update of software in the relationship between supplier and OEM? (Figure 10)

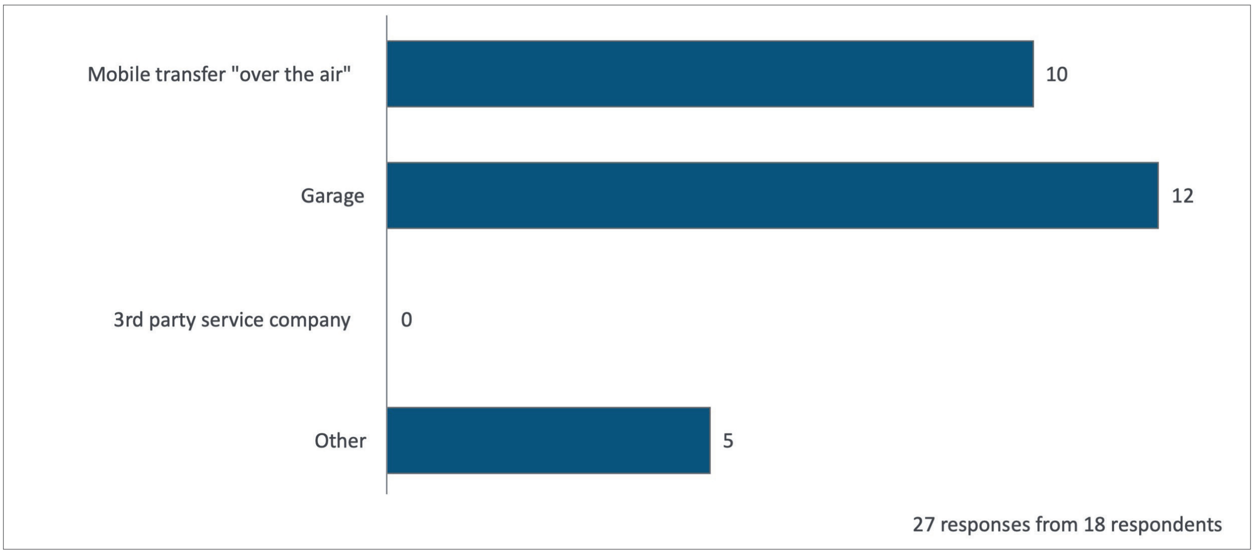


Figure 9: How is the update of software done (multiple choice)?

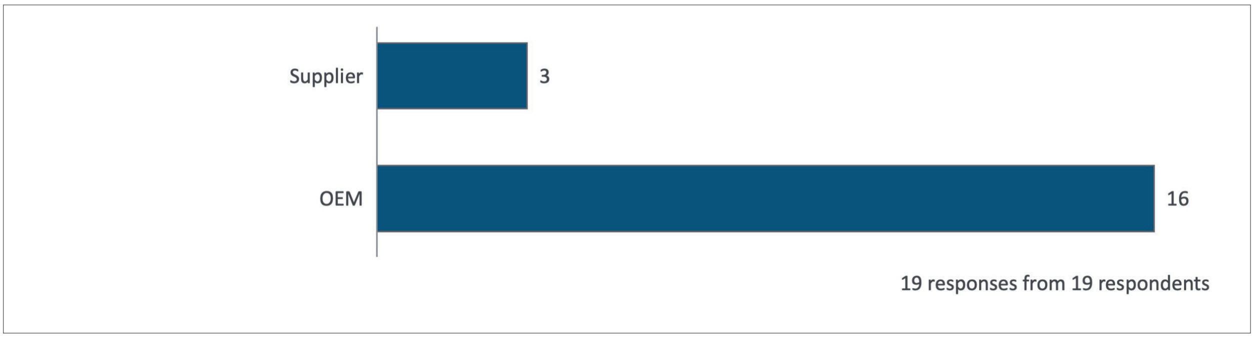


Figure 10: Who is responsible for the update of software in the relationship between supplier and OEM?

When asked how software updates are performed, there were two predefined response options: update "over the air" and update by service staff in the car repair garages. As shown in Figure 9, these two response options were selected by approximately the same number of respondents. This clearly suggests that, in practice, the two options are not available in parallel for specific software. Instead, it must be assumed that either the one or the other option is available for the software. It is probable that embedded software is more likely to be updated by trained staff in car repair garages, while infotainment and navigation software are better suited to over-the-air updates. However, the survey results do not indicate which option is used to update which software. The survey did not ask the participants for details on this type of background information.

The survey results regarding the question of who is responsible for software updates shown in Figure 10 are interesting in that the expertise required to create a software update is more likely to be found at the software suppliers if the software involved is commercial purchased software. A distinction between deciding whether and when an update is necessary for security reasons and the actual creation of an update, on the one hand, and the distribution of a created update, on the other, could probably have been made here. It is clear that the distribution of the update is the responsibility of the OEM due to the fact that the end customer has a purchase or user contract for the product with the OEM. It is not possible to tell from the diagram in Figure 10 whether the responses, which have been divided into OEM and supplier, reveal a largely consistent view of the responsibilities or whether the assessments diverge. The basic data from the survey results can, however, provide information in this context. According to this data, ten out of eleven responses from suppliers (90%) state that the responsibility lies with the OEMs. This statement is confirmed by six out of eight responses from the OEMs (75%).

3.4 Relationship between software components and product structure

To shed light on the relationship between software components and the product structure, survey participants responded to the following question:

- How are software components assigned to your product bill of materials (BoM)?

There were no predefined response options for this question. The participants were able to respond freely, which meant that a variety of possible solutions were recorded.

Most of the responses indicate directly or indirectly that software components are managed with the support of a database in some type of IT system. In these cases, however, IT systems that have been specifically designed for this task (ALM, SVN, etc.) are used. There is no response that clearly indicates that software components and mechanical/mechatronic components are managed in the same IT system. A large number of responses indicate that software components have an identifying part number and that the system used to manage the software components is linked in some way to the PLM system. This means that software components are also included in the bill of materials (BoM) or in the product structure via their part numbers. However, there are also responses that leave open the question of whether or not such a link exists. Very few responses explicitly indicate that no link between software management and a PLM system exists. In a few cases, there appears to be a correlation between software components and product structure that is created manually or without the support of a database.

3.5 Cloud computing and collaboration

The set of topics involving cloud computing and collaboration is the most extensive in this survey. Survey participants responded to the following questions in the context of these topics:

- Do you use public clouds for collaborative software development with external partners? (Figure 11)
- Which elements of self-developed software are exchanged? (Figure 12)
- How are these elements exchanged?
- Which standards are relevant for you in software development? (e.g. quality standards, exchange standards, communication standards) (Figure 13)
- What challenges are critical?
- What expectations do you have towards software suppliers in your own supply chain?
- How satisfied are you with the fulfillment of these expectations? (Figure 14)
- What major topics do you see to be discussed for cross-enterprise software development and deployment?
- What options do you use to protect your company's know-how in software development?

Some of the questions also aim to identify a cross-enterprise need for action, which is traditionally the focus of the Standardization Strategy Board.

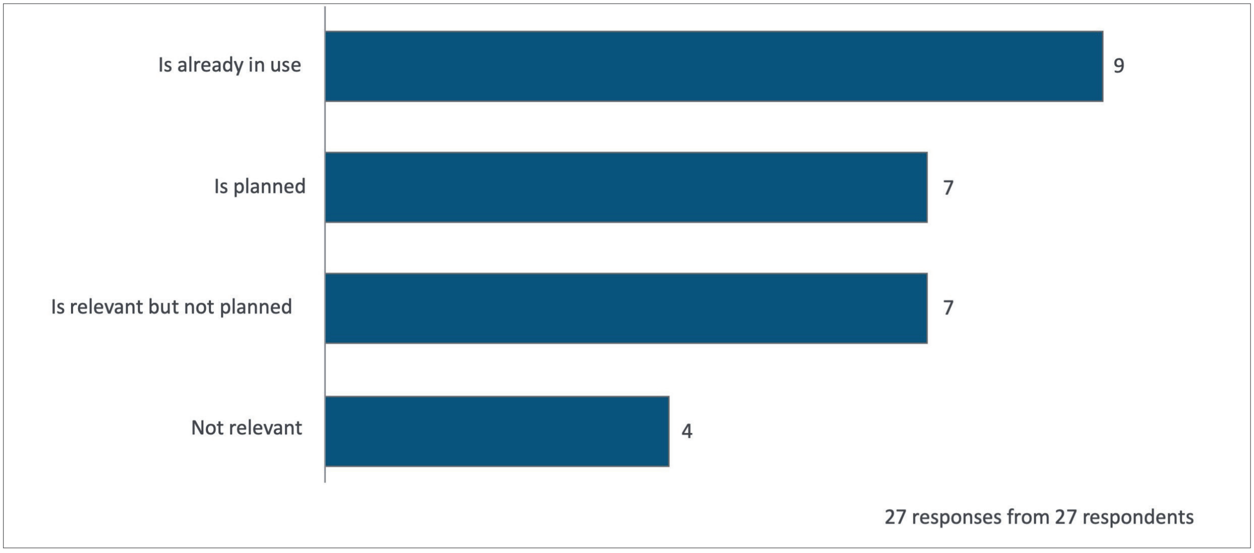


Figure 11: Do you use public clouds for collaborative software development with external partners?

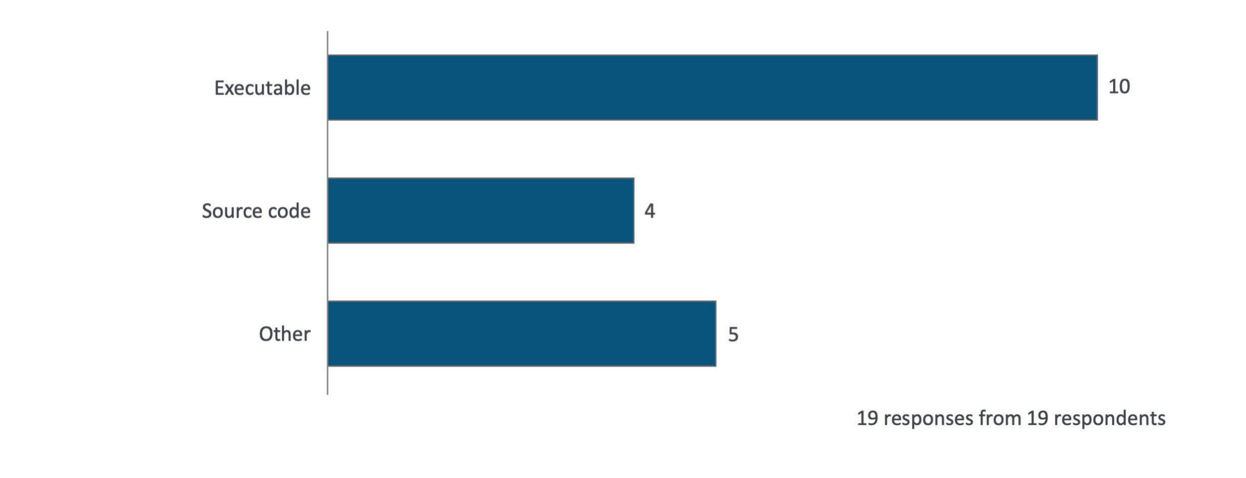


Figure 12: Which elements of self-developed software are exchanged?

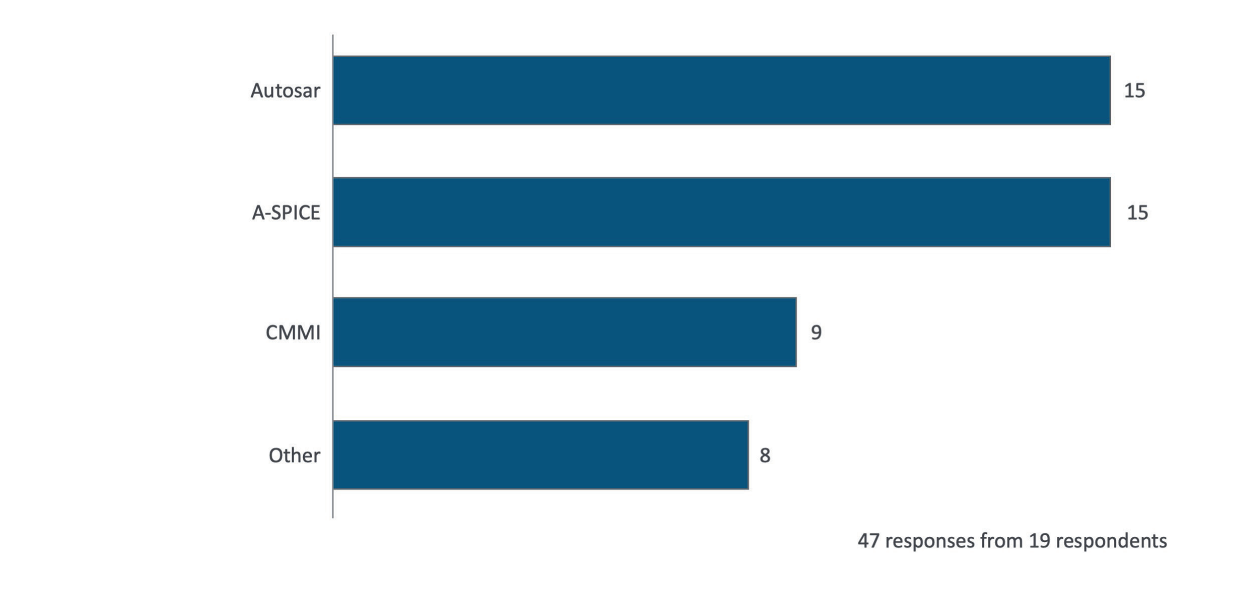


Figure 13: Which standards are relevant for you in software development (multiple choice)? (e.g. quality standards, exchange standards, communication standards)

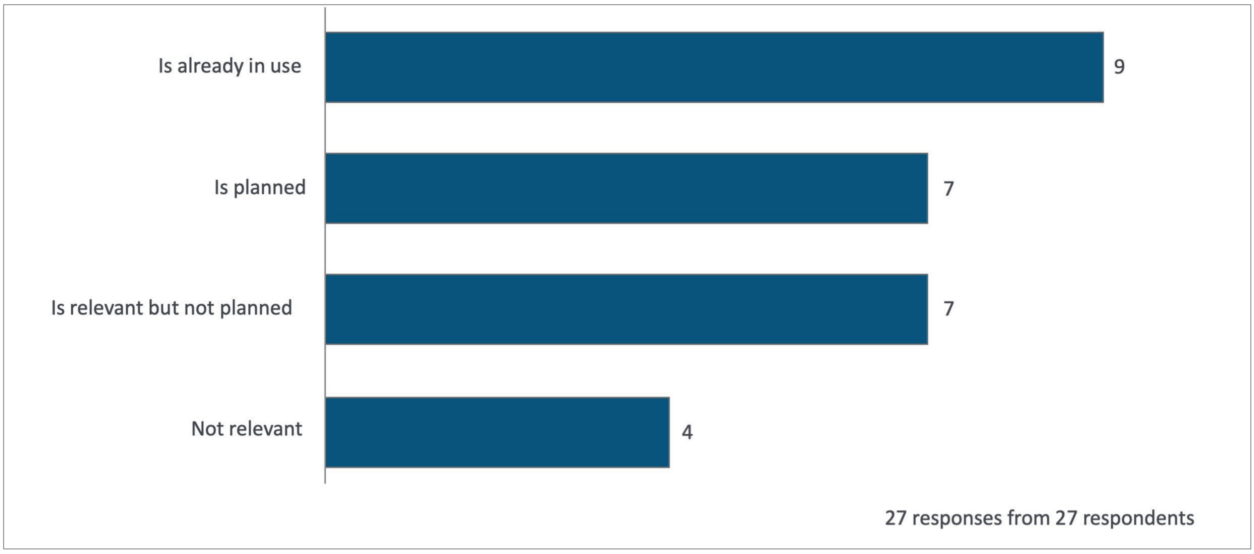


Figure 14: How satisfied are you with the fulfillment of these expectations?

The survey results shown in Figure 11 indicate that over half of the respondents (approx. 60%) are already performing software development in cross-enterprise cloud environments or have concrete plans to do so. About a quarter of respondents said that cross-enterprise software development in the cloud is relevant but no concrete plans for this yet exist.

Figure 12 shows that compiled or executable software components are exchanged in about 50% of cases, while the exchange of source code is mentioned much less frequently. When asked how software components are exchanged, a wide variety of solutions were mentioned, some of them either proprietary or commercial solutions. It was often pointed out that customer-specific systems and solutions exist and that these systems and solutions are to be used. In some cases, encryption was also mentioned in the context of data exchange.

Standards are also of great importance when it comes to cross-enterprise collaboration. Figure 13 shows that AUTOSAR[5] and A-SPICE in particular play a significant role. AUTOSAR focuses on the development of ECU software, while A-SPICE is a process model for improving quality. Companies can also be certified according to A-SPICE. CMMI[6] (Capability Maturity Model Integration) is also important but much less so than A-SPICE. The results do not provide a clear indication of the extent to which the development of infotainment software is also characterized by the use of standards, apart from A-SPICE or CMM. CMMI is a family of reference models for different application areas; these are currently product development, product purchasing and service delivery. A CMMI model is a dependable pool of best practices that helps improve an organization.

The survey results indicating to what extent the expectations placed on the software suppliers are met shown in Figure 14 are worth noting. Although it was often stated that it is difficult to determine whether or not expectations are actually being met, the survey indicates that relatively few respondents are fully satisfied. The somewhat lukewarm response "Rather satisfied" was given most frequently, but "Rather dissatisfied" was also specified relatively often. The expectations placed on the software suppliers mentioned most frequently were the use of and compliance with standards, e.g. A-SPICE, Autosar and TISAX[7], and proof of the requisite expertise and certifications, which can also play a role in the quoting and ordering processes. TISAX (Trusted Information Security Assessment Exchange) was developed by the VDA and involves the secure processing of confidential information, the protection of prototypes and data protection in accordance with the General Data Protection Regulation (GDPR) for potential business transactions between carmakers and their service providers or suppliers. Another key focal point was a general call for compliance with quality criteria as well as a reference to agile development methods.

[5] Autosar
[6] Capability Maturity Model Integration
[7] TISAX

When asked about the options used to protect knowledge, the responses involve predominantly organizational approaches such as the need-to-know principle or other organizational regulations such as non-disclosure agreements. Again, the responses made reference to standards such as TISAX, ISO 21434[8] and ISO 62443[9]. ISO 21434 "Road vehicles – Cybersecurity engineering" is a standard for cybersecurity in motor vehicles. ISO 62443 /IEC 62443 is an international series of standards for "Industrial communication networks – Network and system security". Nevertheless, there are also indications that knowledge is protected by the use of technical options (concealing know-how and encryption). More precise information on the technical solutions used is, however, not available.

The responses to the question regarding critical challenges were very extensive and also very diverse. The topics data security, knowledge protection in the broadest sense and access rights management represent a prominent group of topics. Another key focal point is the integration of relevant IT systems (PLM-ALM integration). A third key focal point involves topics such as complexity, variability, versioning mechanisms and software quality. Regulatory requirements, such as UNECE[10] regulations for example, emerged as a fourth focal point. UNECE is an important set of regulations developed by the United Nations Economic Commission for Europe (UNECE or UN/ECE) with the aim of reducing the technical obstacles that hinder international trade. Specifically, it involves an agreement on the adoption of uniform provisions regarding the approval of equipment for and components of motor vehicles and the mutual recognition of such approval.

When asked about the topics for which the respondents see a need for discussion, there are no clear focal points. The topics traceability, assignment of responsibilities, and interfaces between the IT systems involved were, however, mentioned frequently. Coding guidelines and cloud technologies were also mentioned. Reusable libraries and components and the resulting dependencies are topics for which a need for discussion is seen. This means that these needs tie in, to a certain extent, with the critical challenges mentioned earlier.

[8] ISO/SAE 21434
[9] ISO 62443 / IEC 62443
[10] UNECE

4 Summary and outlook

The current status of software development in the automotive industry was surveyed using a total of 26 questions involving five groups of topics and is documented in this white paper.

- Use of free and open-source software
- Handling security vulnerabilities and potential risks
- Update strategies and software maintenance
- Relationship between software components and product structure
- Cloud computing and collaboration

The results of the survey certainly highlight potential for improvement. For example, it is apparent that OEMs and suppliers have, to a certain extent, different expectations when it comes to response times in the event of software errors or security vulnerabilities. This is indicated in the diagram in Figure 8. This suggests that the cause is more likely communicative in nature. This assumption could be explored in more detail. Be that as it may, we can say that the sufficient and precise communication of expectations regarding response times in the event of software errors is an important means of improving quality.

The same applies to general expectations regarding the quality of products and processes. The question of whether the OEMs' general expectations regarding the product and process quality provided by the software suppliers are actually being met does not appear to be an easy one to assess from the OEMs' perspective. The survey results indicate that not all expectations are being met. In this case, it would be important to determine the cause. It might be possible to determine whether the communication of expectations and the tracking of the extent to which these expectations are met could be improved.

Software development in cross-enterprise clouds poses a number of challenges, which primarily involve the management of access rights, knowledge protection and data security. Traceability and the handling of the complexity and variability of software products is also a major challenge that needs to be tackled. Taken as a whole, this is a comprehensive topic that could be examined in a cross-enterprise context within the framework of the prostep ivip Association.

The survey participants see further need for discussion on topics such as traceability, assignment of responsibilities and interfaces between the IT systems involved, coding guidelines, cloud technologies and the reusability (reuse) of software components.

Two standards that were not addressed directly in the survey (and are therefore not represented in the survey results) but which have recently gained in importance are the CVE[11] (Common Vulnerabilities and Exposures) and CVSS[12] (Common Vulnerability Scoring System) standards. Common Vulnerabilities and Exposures (CVE) is a referencing system maintained by the U.S. National Cybersecurity FFRDC, which is operated by the Mitre Corporation. The objective is to establish a uniform naming convention for security vulnerabilities and other vulnerabilities in computer systems. The Common Vulnerability Scoring System, is an industry standard for evaluating the severity of potential or actual security vulnerabilities in computer systems. In the CVSS, security vulnerabilities are evaluated according to different criteria, referred to as "metrics", and compared with each other so that a priority list for countermeasures can be created. These two standards are already being used in industry practice and are worth analyzing in terms of their relevance and applicability for software development and software maintenance in the automotive and aerospace industries. They are therefore being examined and evaluated within the framework of the Standardization Strategy Board.

We would once again like to take this opportunity to thank all those who participated in the survey for their time and commitment, and we hope that we have provided a few ideas and input for further study.

[11] CVE
[12] CVSS

prostep ivip association

Dolivostraße 11
64293 Darmstadt
Germany

Phone +49-6151-9287336
Fax +49-6151-9287326

psev@prostep.com
www.prostep.org

ISBN 978-3-948988-20-3
Version 1.0, 2022-5