Journal of Autonomous Vehicles and Systems. Received February 07, 2026;
Accepted manuscript posted May 8, 2026. doi:10.1115/1.4071897

A synoptic comparative analysis of aviation and automotive software development processes along the V-model to deduct transfer proposals for autonomous driving

Yusuf Akkus
Daimler Truck AG
Stuttgart, Germany
Email: yusuf.akkus@daimlertruck.com

Björn Annighöfer
Professor at University of Stuttgart
Institute of Aircraft Systems
Stuttgart, Germany
Email: bjoern.annighoefer@ils.uni-stuttgart.de

The rapid growth of software complexity in autonomous driving systems poses significant challenges for ensuring functional safety in automotive development. Aviation has historically achieved higher safety levels through mature development processes, structured assurance practices, and strong regulatory involvement. This paper presents a comparative analysis of automotive and aviation software development methods along the V-model, with the aim of identifying concrete transfer potentials from aviation to automotive development. The analysis reveals that the most significant benefits arise at the system and process governance levels, particularly through early and continuous authority involvement and structured system assurance practices. Further transferable elements are identified in software requirements engineering, tool qualification, and development planning, while coding standards represent the only lifecycle stage where automotive development is more prescriptive. The results indicate that selectively adopting aviation-derived practices can strengthen confidence in the safety of autonomous driving systems. Overall, the study demonstrates that a targeted transfer of aviation development principles offers a viable path toward higher safety assurance for future automotive technologies.

1 INTRODUCTION

The automotive industry has undergone enormous transformation with disruptive trends such as autonomous driving, electrification, and connectivity [1]. In recent years, the integration of AI (artificial intelligence) has also emerged as a key focus area. Consequently, development complexity has increased, fueled by the increasing number of E/E (electric and electronic) devices [2] and sophisticated SW (software) components surpassing aviation, e.g. in the metric lines of code with more than 100 million lines of code [3]. The industry is shifting towards SW-centric vehicles and advanced architectural solutions, with many companies developing proprietary operating systems [4]. These innovations introduce substantial challenges, particularly in maintaining system safety and reliability [5], which remains an ongoing concern. This is largely due to the inherent complexity of these technologies, necessitating a systematic and well-structured approach. In contrast, aviation has decades of experience managing complex electronic systems, including fly-by-wire technology and high levels of automation [6]. On the development side, discipline and rigorous safety culture have fostered strong confidence in system reliability and an exceptional safety record. Figures 1 and 2 highlight aviation's safety advantage, illustrating death rates in absolute numbers and per distance traveled, and overall fatalities across various modes of transportation. Akkus and Annighöfer concluded that the safety record of aviation is due to its early standardization and rigorous systematic development approach [7], highlighting that carrying over development methods from aviation to automotive has potential benefits.

1.1 Background

According to [7] several areas with transfer potential were identified, such as authority involvement and common cause analysis. Building on this, a detailed analysis

Journal of Autonomous Vehicles and Systems. Received February 07, 2026;
Accepted manuscript posted May 8, 2026. doi:10.1115/1.4071897

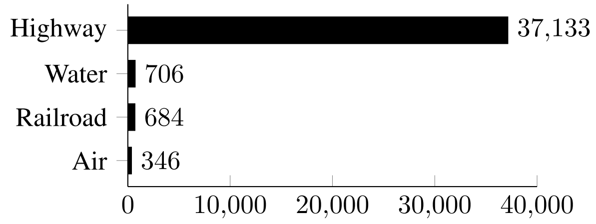


Fig. 1. Absolute death numbers by transportation mode in United States in 2021 [8]

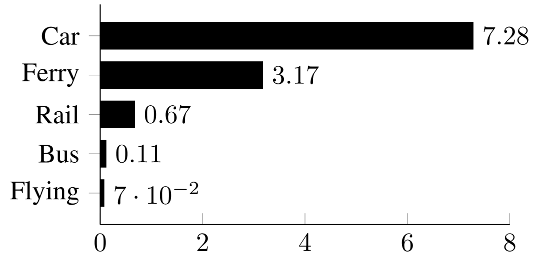


Fig. 2. Fatalities per 1 billion passenger miles, US 2000-2009 [9]

of development methods has been conducted. Whereas the focus of this work is the comparison of both areas, the transfer direction is only towards automotive. Also, we intend to add benefit only for new trends like autonomous driving and not for established state-of-the-art automotive development. Due to that, the following research questions are defined in the next subchapter

1.2 Research questions

As the comparison rationale is evident and rough direction is defined in this subchapter research questions (RQs) will be listed.

RQ1: How can aviation and automotive be compared? **Rationale** This question implies both system and software. Due to differences of both areas and the fact that comparing them has a very broad scope, a proper focus is to be defined.
RQ2: What can be transferred from aviation to automotive? **Rationale** Which topics (process, methods) are beneficial for a transfer to automotive, especially for the development of emerging technologies?
RQ3: What is the quantified benefit of each transfer proposal? **Rationale** Each transfer proposal shall be assessed in terms of its benefit in a quantified way in order to provide a precise indication whether to adopt or not.

2 RELATED WORK

In this section related work will be presented in order to identify the research gap and rational.

Based on the research questions and the existing related work, the following research gap has been identified. In [7], an initial direction for transferring methods from aviation to automotive was outlined. Other studies highlight specific aspects: for example, that ISO26262 can benefit from DO-178B [10], that autonomous technologies require more rigorous methods [11], that development effort for COTS hardware could be reduced if ISO26262 applied the same rigor as DO-254 [12], and that a mapping of rigor for development artifacts can be established [13]. However, these works remain largely comparative and do not provide concrete transfer proposals. Baek et al. [14] attempt to transfer methods by discussing potential benefits, but their focus is on transferring practices from automotive to aviation and does not address autonomous technologies. This reveals a clear gap: a comprehensive comparative analysis with concrete, quantitatively supported transfer proposals from aviation to automotive, specifically for autonomous driving, is still missing. With regard to software architecture, Leiner [15] investigates integrated architectural approaches across different domains, focusing primarily on partitioning strategies. Gaska [16] examines the challenges of next-generation Integrated Modular Avionics (IMA) concepts, and later [17] proposes a model-based engineering approach for the aviation IMA domain, suggesting that similar techniques could benefit the automotive IMA domain. Bandur [18] concentrates mainly on the centralization of automotive E/E architectures, but also provides a comparison with aviation, noting that the adoption of IMA principles in automotive represents a successful technology transfer. Overview is shown in Table 1

3 COMPARISON METHOD

Several topics with significant transfer potential were identified in previous works, including common-cause analysis for autonomous driving, enhanced involvement of regulatory authorities, complexity management through rigorous software development processes, and tool qualification. These findings are noteworthy because they align with the well-established development framework of the V-model. This implies that a systematic comparison of development methods and processes across both domains, structured along the V-model, ensures that the areas relevant to each transfer proposal are consistently identified. Consequently, choosing the V-model as the comparison pattern provides an effective means of

Journal of Autonomous Vehicles and Systems. Received February 07, 2026;
Accepted manuscript posted May 8, 2026. doi:10.1115/1.4071897

Table 1. Comparison of the State of Research

Work	***Content***	***Conclusion***
[7]	Comparative analysis of aviation and automotive methods	Justification of comparison and identification of transferable development methods
[10]	Mapping of safety standards	ISO 26262 can benefit from DO-178B through mapped rigor levels
[11]	Comparison of software development safety standards	Emphasis on appropriate methodologies for autonomous systems
[12]	Comparison of hardware development standards	Design assurance effort may be reduced if DO-254 rigor is applied
[13]	Comparison of design assurance levels	Cross-industry rigor mapping based on development artefacts
[14]	Automotive vs. aviation software development	Potential for method transfer from automotive powertrain development
[15]	Partitioning of integrated systems	Maturity gaps identified between operating systems
[16]	Architecture centralization approaches	Need for distinct data and architecture infrastructures identified
[17]	Model-based engineering comparison	Aviation maturity highlights automotive development gaps
[18]	Automotive E/E architectures	Centralized aviation architectures proposed for automotive

capturing the essential aspects of development, as normative standards are themselves organized around this model [19]. Furthermore, aviation's highly regulated development environment [14], [10] necessitates strict adherence to the V-model, with all major standards oriented toward it. Thus, the methodological comparison between aviation and automotive development is largely predefined: follow the V-model and examine the corresponding normative guidance. In concrete terms, the comparison methodology proceeds according to the sequence of development stages defined in the V-model. The analysis begins at the system level, including system-safety considerations, followed by the stages of software requirements and software architecture. Subsequently, software coding and implementation are examined, and the comparison concludes with verification. For each stage, the first step is a direct comparison of the relevant normative guidance, primarily ISO26262 versus DO-178C (including ARP-4754 and ARP-4761). We evaluate the differences and assess their implications for the safety record. If a method appears potentially beneficial, we further analyze the feasibility of transferring it to the automotive domain, taking into account all industrial boundary conditions and explicitly examining whether and how such a transfer would be possible. Once adapted to the constraints of automotive development, a concrete transfer proposal is formulated. The second step in each stage involves comparing industry best practices. This includes examining aspects outside the scope of normative guidance, focusing on common challenges and on how these are addressed in both domains. Scientific literature emphasizes that normative standards alone are insufficient to fully characterize how safety-critical systems are developed in practice, as standards prescribe required processes but do not reflect the actual conduct of engineering work in industrial environments. Studies in safety engineering distinguish between prescribed and observed practices and argue that meaningful analysis must incorporate empirical observation of real industrial behavior, not only normative compliance [20]. Likewise, [21] highlights the importance of experience-based development. Furthermore, industrial surveys of safety-critical software development show that verification, validation, and testing activities are frequently adapted or extended beyond

Journal of Autonomous Vehicles and Systems. Received February 07, 2026;
Accepted manuscript posted May 8, 2026. doi:10.1115/1.4071897

what standards formally specify, demonstrating the importance of understanding best practices as applied in real projects [22]. Empirical studies confirm that organizational culture and industrial practice influence how standards are interpreted and applied, reinforcing the need to complement standard-based analysis with evidence from practice when comparing aviation and automotive across lifecycle models such as the V-model [23]. The chronology is shown in Figure 3.

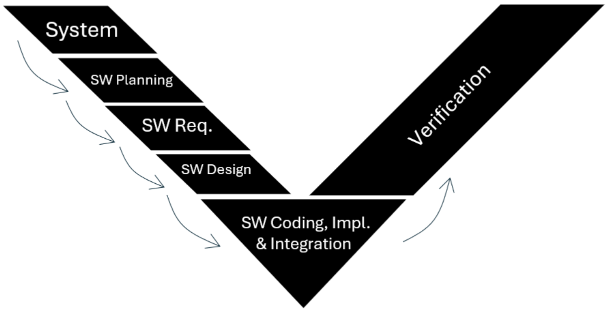


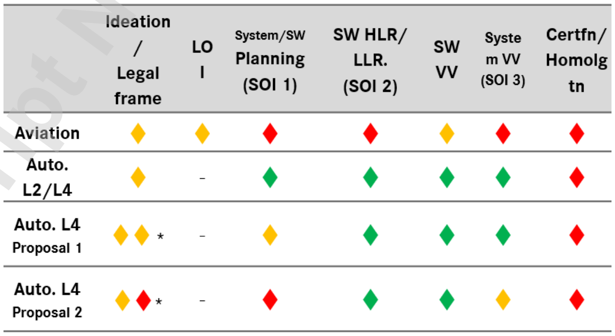
Fig. 3. Comparison chronology along the V-model

4 COMPARISON

The comparison presented in this chapter is dedicated to the challenges introduced by autonomous driving systems. Compared to conventional driver assistance functions, autonomous systems exhibit significantly increased functional complexity, stronger interdependencies between perception, decision, and actuation layers, and stricter safety requirements. These characteristics make the robustness, traceability, and verifiability of development processes a critical factor when assessing the suitability of existing practices for future automotive applications. A synoptic and comparative overview of the key findings is presented and it is examined whether, and to what extent, specific processes and methods can be transferred from aviation to emerging automotive technologies. The structure of the subchapters follows the V-model, in alignment with the normative guidance of DO-178C [24] and ISO 26262 [25]. The discussion begins at the system level, as this forms the foundation of software development [26], and then follows the order described in Figure 3.

4.1 System level

System-level development exemplifies the strong normative and regulatory guidance in aviation. At this level, the comparative investigation shows that intensive involvement of authorities (FAA in the U.S. or EASA in Europe) has contributed to confident development processes and safer products. With all mandatory stages of involvement (SOIs) and certification steps, aviation companies meet with authorities at least five times during a project [27]. By contrast, automotive performs such an interaction only once during homologation; in the U.S., even the homologation process does not require authority involvement [28]. While this approach has been adequate for legacy automotive systems, it is evident that complex technologies such as autonomous driving require stricter authority involvement [7]. Accordingly, we analyzed how authority involvement could be increased for autonomous driving, considering the relevant know-how of authorities. Two proposals were ultimately derived: a soft option recommending an increased number of meetings with authorities, and a more rigorous option requiring at least one mandatory step immediately after system and software planning, corresponding to SOI 1 in aviation, shown in Figure 4.



	Ideation / Legal frame	LOI	System/SW Planning (SOI 1)	SW HLR/ LLR. (SOI 2)	SW VV	System VV (SOI 3)	Certfn/ Homolgtn
Aviation	🟧	🟧	🟥	🟥	🟧	🟥	🟥
Auto. L2/L4	🟧	–	🟩	🟩	🟩	🟩	🟥
Auto. L4 Proposal 1	🟧🟧 *	–	🟧	🟩	🟩	🟩	🟥
Auto. L4 Proposal 2	🟧🟥 *	–	🟥	🟩	🟩	🟧	🟥

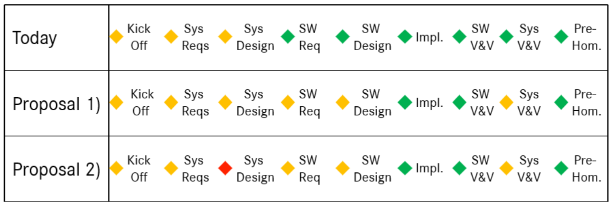
* The number of diamonds does not represent the number of meetings with authorities. The colors indicate whether a meeting takes place (red, orange) or not (green, "–"). Two diamonds in the same cell mean that, according to our proposal, the interactions are increased, without specifying a concrete number.

Fig. 4. Authority involvement enhancement proposals

While strong normative guidance and authority involvement ensure process assurance in aviation, autonomous driving may similarly benefit from additional process oversight. ASPICE (Automotive Software Process Improvement and Capability dEtermination) is a widely used process audit instrument [29]. OEMs typically conduct this audit once during a project, mainly at the beginning, to assess supplier process readiness. This audit mechanism can be leveraged to enhance process assurance in autonomous-vehicle projects. Unlike author-

Journal of Autonomous Vehicles and Systems. Received February 07, 2026;
Accepted manuscript posted May 8, 2026. doi:10.1115/1.4071897

ities, ASPICE assessors must demonstrate deep technical domain knowledge [30], making their involvement a valuable complement to authority oversight at the system-design and software levels. Here as well, two proposals were derived: a soft option recommending an increased number of assessments, and a rigorous option requiring at least one mandatory assessment during the system-design phase, shown in Figure 5.



Today	🟧 Kick Off	🟧 Sys Reqs	🟧 Sys Design	🟩 SW Req	🟩 SW Design	🟩 Impl.	🟩 SW V&V	🟩 Sys V&V	🟩 Pre-Hom.
Proposal 1)	🟧 Kick Off	🟧 Sys Reqs	🟧 Sys Design	🟧 SW Req	🟧 SW Design	🟩 Impl.	🟩 SW V&V	🟧 Sys V&V	🟩 Pre-Hom.
Proposal 2)	🟧 Kick Off	🟧 Sys Reqs	🟥 Sys Design	🟧 SW Req	🟧 SW Design	🟩 Impl.	🟩 SW V&V	🟧 Sys V&V	🟩 Pre-Hom.

* The colors indicate whether an assessment should take place as homologation-relevant (red), recommended (orange), or not at all (green).

Fig. 5. ASPICE enhancement proposals

During the system-level safety comparison, the strong emphasis on common-cause analysis in aviation was also identified as a potential benefit for autonomous systems [7]. Due to fail-operational architectures and redundancy concepts, common-cause analysis methods have long been state of the art in aviation. They are referenced in ARP4754 [31] and described in detail in ARP4761 [32]. With autonomous driving, automotive will deploy redundancy in safety-critical systems for the first time, for example in braking and steering systems [33]. In [33], multiple failure scenarios were classified and categorized according to their impact in order to focus the analysis. In a second step, an identification method for dependent failures was developed using the aviation common-cause approach. A list of potential dependent failures was generated together with questionnaires (Figure 6). Results showed a perfect match with a parallel supplier-side analysis, which supports the proposal to apply this common-cause identification methodology in system-level safety analysis.

4.2 Software planning

The analysis of the software planning stage shows a very stringent documentation effort on the aviation side; for example, PSAC (Plan for Software Aspects of Certification), SDP (Software Development Plan), and similar documents are artifacts of this stage. Although the automotive domain also has certain expectations regarding the planning stage, it is by far not as strict as aviation. A counterpart to the PSAC does not exist. The reason aviation requires a PSAC is that immediately after the planning stage, SOI 1 follows as the first mandatory interaction with authorities. Since there is no such authority interaction at this stage in automotive, documentation similar to the PSAC has not been established to date. We recognize the importance of increasing the stringency of planning artifacts for autonomous systems [7]. As planning artifacts are assessed during SOI 1, at this stage we refer to our proposal on enhanced authority involvement, as it addresses precisely this gap.

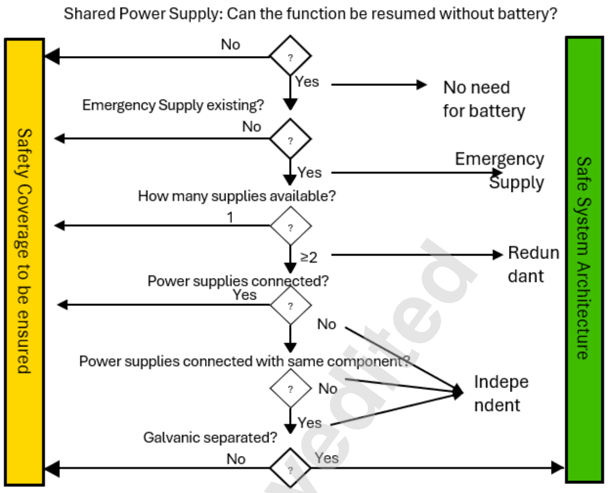


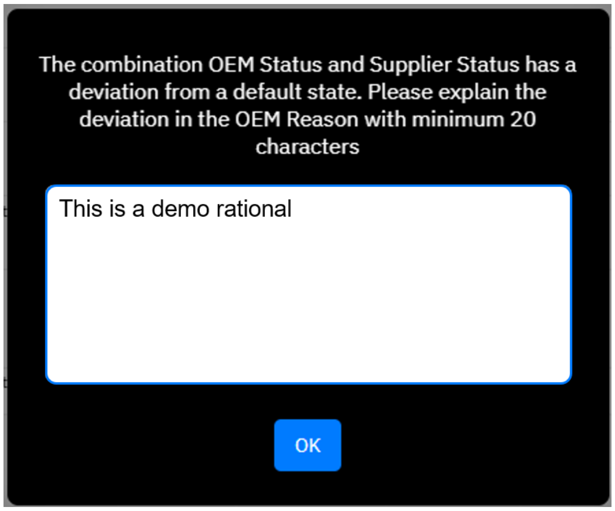
Fig. 6. Example for a questionnaire of the identification model

4.3 Software requirements

The analysis shows that a strict separation of high-level requirements (HLR) and low-level requirements (LLR) helps establish better traceability and a design-independent specification [26]. In contrast, automotive lacks this two-level hierarchy in ISO 26262 Part 6, although it implicitly integrates additional requirement hierarchies [34], [35]. Furthermore, ASPICE decomposes software engineering modules into three categories [36], clearly separating software requirements from design. We propose establishing a mandatory two-level requirements hierarchy. In addition, we observe that the automotive industry is currently increasing its activities toward systems-engineering approaches, requiring clear separation between requirements and design [37], [38], [39]. Further normative guidance comes from the FAA (Federal Aviation Administration) requirements handbook [40], which explicitly states that documentation of requirements rationale must be enforced and provides further explanation on how this should be done.

Journal of Autonomous Vehicles and Systems. Received February 07, 2026;
Accepted manuscript posted May 8, 2026. doi:10.1115/1.4071897

We analyzed this for automotive settings and developed a forcing concept on a commercial engineering lifecycle platform (tool-based), enforcing the documentation of rationale whenever a deviation from a default alignment state is recorded; see the example in Figure 7. To leave the artifact, an entry of at least 20 characters must be provided, which activates the “Ok” button. On the same platform, another concept, conditioning, is proposed. This approach automates the setting of certain requirement attributes to predefined states in order to reduce human error; for example, changed requirements states are automatically reset, ensuring that reviews for each change are not missed. Our final proposal for requirements engineering originates from industry best practices. In [26], the use of prototypes for requirements engineering in aviation is recommended. Early validation is common in aviation through model-based engineering [41], [42]. The automotive industry has likewise recognized the importance of early validation and applies the so-called shift-left approach to verify products at early stages [43], [44], [45]. Our specific proposal is to intensify early validation.


The combination OEM Status and Supplier Status has a deviation from a default state. Please explain the deviation in the OEM Reason with minimum 20 characters

This is a demo rational

Fig. 7. Tool-based rationale documentation of requirements with deviating alignment status

We highlight the increasing relevance of these software requirements practices for autonomous driving systems. Autonomous functions rely on complex software architectures and extensive interaction between safety-critical components, which enhances the need for precise requirements decomposition, bidirectional traceability, and rigorous review procedures. The more formalized requirements hierarchy and verification culture that has traditionally been applied in aviation therefore provide mechanisms that can help manage the higher system complexity and safety assurance demands associated with autonomous vehicle development.

4.4 Software design

At the software design level, normative guidance in both industries does not differ significantly [24], [25]. Although strong cohesion and loose coupling are emphasized in both domains, we did not identify substantial differences on the normative guidance side. Our analysis therefore focused on industry best practices, concluding that the main trend is the increasing demand for high-performance computing driven by data-intensive applications such as AI (artificial intelligence) [46], [47], [48]. Software-defined vehicles (SDVs) are a prominent automotive example of such data-intensive applications. This trend drives architectural evolution, for example toward centralization approaches, as a response to the growing computational demands [49]. Aviation has also adopted centralization concepts due to space, power, and weight constraints, hosting a high number of functions using the so-called IMA (Integrated Modular Avionics) platform [50]. Data-intensive applications are also an emerging trend in aviation.

4.5 Software coding, implementation, and integration

At the coding level, an interesting observation emerges. Unlike in any other stage of the V-model, the prescriptiveness of normative guidance in the automotive domain exceeds that of aviation. While DO-178C objectives do not mandate a specific coding standard, ISO 26262 explicitly and strongly recommends MISRA C [51] as a coding standard that satisfies ISO requirements. This reflects the preventive, rule-based measures needed to mitigate software risks in complex, supplier-driven environments. Established guidelines are crucial for eliminating unsafe language constructs early in development. Regarding language choice, both domains offer limited options due to prescriptive rules, legacy constraints, and compliance objectives [52], [53]. Consequently, no method transfer is proposed here. During software coding, implementation, and integration, particular attention is given to tools themselves. A notable difference emerges when comparing tool qualification processes. DO-178C addresses tool qualification within its integral processes; however, the detailed and rigorous qualification procedure is provided in a separate document, DO-330 [54], which closely parallels the structure of DO-178C. In contrast, ISO 26262 Part 8 dedicates only a few pages to the topic, outlining minimal objectives

Journal of Autonomous Vehicles and Systems. Received February 07, 2026;
Accepted manuscript posted May 8, 2026. doi:10.1115/1.4071897

without prescribing a specific approach. It also acknowledges that tool development compliant with other standards, such as DO-178C (and by implication DO-330), is acceptable. The automotive domain recognizes the growing importance of tool qualification due to the increasing number of software components and requirements [55], [56]. We propose applying DO-330-style tool qualification for autonomous driving and SDVs, both for development and verification tools. This implies following the DO-330 development structure and fulfilling its objectives; see Figure 8.

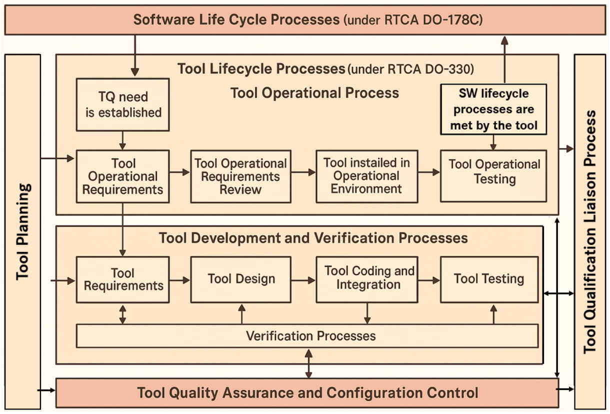


Fig. 8. DO-330 style tool qualification process [57]

4.6 Software verification

Standards for verification in the aviation and automotive domains reveal notable differences in prescriptiveness and flexibility. Aviation adopts an objective-based verification philosophy with strict guidance, whereas automotive standards allow greater flexibility within their normative frameworks. Neither domain mandates static analysis, but aviation places strong emphasis on requirement coverage and traceability, while automotive primarily requires traceability. Coverage criteria also differ significantly: aviation mandates Modified Condition/Decision Coverage (MC/DC) for higher DALs, whereas automotive recommends MC/DC but accepts justification with rationale when full compliance cannot be achieved. We anticipate the most significant divergence in practical implementation, particularly regarding metrics-based assessment of verification techniques and methods. The verification path within the V-model encompasses a broad scope and differs substantially from the development side. While the left side of the V-model begins similarly for both aviation and automotive, starting from an initial state with no existing product, the right side introduces domain-specific variations. At this stage, verification focuses on an already developed product, making the process inherently dependent on domain requirements. Consequently, the comparison strategy centers on cost and efficiency, as verification exerts a substantial influence on overall effort and budget. Both domains allocate approximately 30–40% of total costs to verification, indicating no major difference in expenditure. The second aspect examined is verification efficiency, which can be quantified through metrics [58], commonly used to assess reliability and safety [59]. Comparing both domains based on such metrics aims to determine whether effective practices from aviation can be transferred to automotive applications. Metrics such as defect detection efficiency, defect removal efficiency, and test case effectiveness were selected for this purpose.

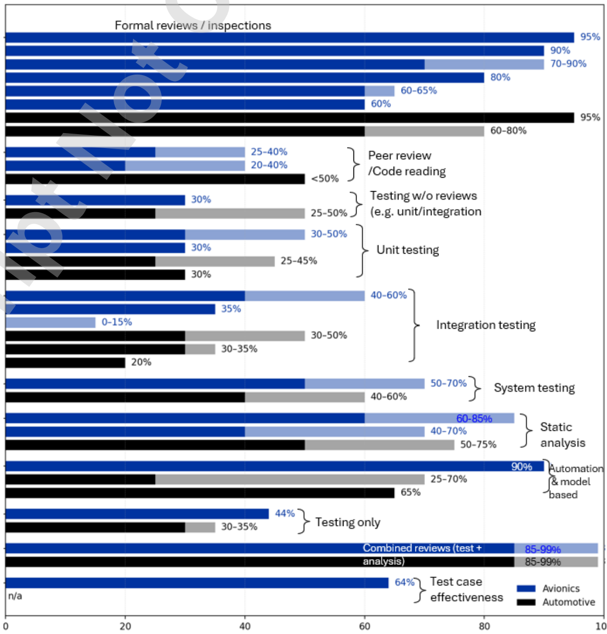


Fig. 9. Overview of software metrics comparison

Figure 9 shows values for each submetric. The values are gathered with a literature review to have comparable data about the above-mentioned metrics to compare software development from both domains. We observe that for each submetric, the values from both domains are similar to each other. Investigations did not spot a significant difference that could be a source for potential transfer proposal.

Further analysis of the verification stage focused on independence, a concept that is called IV&V (indepen-

Journal of Autonomous Vehicles and Systems. Received February 07, 2026;
Accepted manuscript posted May 8, 2026. doi:10.1115/1.4071897

dent verification and validation) and strongly emphasized in aviation [26]. Normative guidance in both domains shows no significant differences, and industry practices largely follow these guidelines. The primary distinction lies in the strong involvement of certification authorities in aviation, which introduces an additional layer of independence. The structural distinction forms the basis for the transfer proposal concerning increased authority involvement discussed in earlier sections. This approach could also be adopted in the automotive domain. However, instead of generating additional redundant proposals, we believe that previously mentioned proposal authority involvement covers the need of the additional independence as well. This is why here we refer to the enhanced authority involvement proposed in the system level.

4.7 Summary of proposals

Table 2 provides the summary of potential transfer proposals from aviation to automotive.

Table 2. Summary of transfer proposals

V-model Stage	*Proposal*
System	Recommended enhanced authority involvement
System	Mandatory enhanced authority involvement
System	Recommended ASPICE enhancement
System	Mandatory ASPICE enhancement
System	Dependent failure analysis
SW requirements	Two-level hierarchy
SW requirements	Rational documentation forcing
SW requirements	Requirements attributes conditioning
SW requirements	Prototyping
SW coding, implementation	Tool qualification

5 QUANTIFIED FEASIBILITY

To evaluate the feasibility of each proposal, a quantification method was developed [60]. This method is based on the project management triangle, which is commonly used in decision-making for automotive projects [61]. The overall estimated feasibility $Score_{Estimated}$ for each proposal is calculated using two main parameters: 1) The $Impact_i$ on the constraints of cost, quality, and time; 2) The $Weighting_i$ of each constraint, derived from an employee survey, as shown in Equation 1.

$$Score_{Estimated} = \frac{\sum_i Weighting_i \cdot Impact_i}{\sum_i Weighting_i} \tag{1}$$

In Equation 1, the index i represents the number of the constraints: cost, quality, and time, which makes i=3.

5.1 Impact Assessment

$Impact_i$ is determined through expert evaluations based on experience in the relevant domain. Impact assessment requires substantial domain expertise, limiting the number of qualified evaluators; therefore, expert interviews are most suitable. Following Hove et al.'s guidance for rigorous empirical interviews [62]. Each impact is rated on a scale from 0 to 10. For every proposal, at least three domain experts were consulted. The assessment process consisted of three steps: 1) Explanation of the proposal; 2) Explanation of the assessment method; 3) Execution of the assessment.

5.2 Weighting Calculation

$Weighting_i$ is derived from an internal employee survey conducted within Daimler Truck [63], with 126 participants. Determining the weighting factor requires a more representative sample to reflect broader tendencies in automotive development. Unlike impact assessment, this stage demands larger sample sizes, as noted by Runeson et al. [64]. Surveys are well suited for this purpose in empirical software engineering. Lim [65] emphasizes their importance in automotive organizations, and Molleri et al. provide a detailed checklist for conducting high-quality surveys [66]. Following their guidance, our study implemented a structured process including research planning, target population identification, survey design and validation, participant recruitment, response management, and data analysis. The survey addressed two groups of questions: 1) Demographic questions, enabling focus on responses from specific expertise areas depending on the proposal; 2) Preference questions, assessing how participants prioritize the constraints of cost,

Journal of Autonomous Vehicles and Systems. Received February 07, 2026;
Accepted manuscript posted May 8, 2026. doi:10.1115/1.4071897

quality, and time relative to each other. This distinction will help tag each proposal depending on which stage of V-model it relates to or can exact transfer is intended (process/method or tool).

5.3 Validation of quantification feasibility method

The proposal about dependent failure analysis was already applied to the safety engineering during development, which provides the chance for a validation. As a result, 14% estimation accuracy was achieved, which classifies the method as ”good” according to basic mathematics [67] and ”low error” according to Project Management Handbook [68].

5.4 Results of quantified feasibility of proposals

Table 3 summarizes the results of impact assessments as well as weighting calculation for each proposal and the overall feasibility score.

Table 3 presents the weighting and impact values for the constraints: cost, quality, and time, in that order. The weighting factors originate from an employee survey with applied tags, while the impact values are based on expert assessments conducted in 2025 and 2026. The rightmost column displays the overall (forecasted) feasibility score. This score ranges from 0 to 10, where 0 indicates the lowest feasibility and 10 the highest. A score of 5 represents a neutral long-term feasibility level.

Figure 10 provides a synoptic overview of the transfer proposals. It reveals that the left side of the V-model demonstrates higher potential, with tool qualification achieving the highest score, and the proposed requirements engineering methods. Also, for system-level development as well as safety methods there is a high potential for transfer possibilities. In terms of prescriptiveness, it is clear that aviation dominates automotive, with the sole exception being coding standards, where automotive explicitly urges the use of MISRA C.

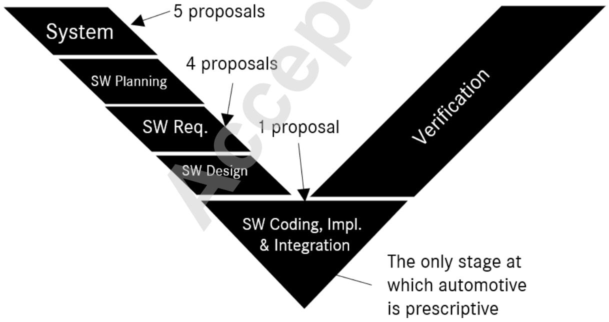


Fig. 10. Transfer potentials across the V-model

Table 3. Summary of overall feasibility calculations for transfer proposals

Proposal	***Weighting$_i$***	***Impact$_i$***	***Overall Feasibility Score***
Recommended enhanced authority involvement	2.59, 3.76, 3.364	5.66, 6, 5.66	5.79
Mandatory enhanced authority involvement	2.59, 3.76, 3.364	3.66, 6, 3.66	4.54
Recommended ASPICE enhancement	2.59, 3.76, 3.364	6, 7.5, 5.5	6.38
Mandatory ASPICE enhancement	2.59, 3.76, 3.364	5, 8, 5.5	6.31
Dependent failure analysis	-*	-*	5.83
Two-level hierarchy	2.57, 3.74, 3.68	5.66, 6, 8	6.15
Rational documentation forcing	2.59, 3.76, 3.364	5.33, 7.66, 6.66	6.69
Requirements attributes conditioning	2.59, 3.76, 3.364	5.33, 6.66, 6.66	6.32
Prototyping	2.57, 3.74, 3.68	4.66, 7.33, 6	6.65
Tool qualification	2.59, 3.76, 3.364	9.66, 10, 8.66	9.42

*No values for dependent failure analysis calculated since this method is already applied. Overall value represents actual feasibility, not forecast

The presented quantitative scores are derived from empirical inputs. The numerical values should be interpreted as context-specific and indicative rather than statistically representative for the entire automotive industry. This is why we again highlight that these results origi-

Journal of Autonomous Vehicles and Systems. Received February 07, 2026;
Accepted manuscript posted May 8, 2026. doi:10.1115/1.4071897

nate from a model that consists of two empirical components: *Weighting*$_i$ (W_i) and *Impact*$_i$ (I_i). While *Weighting*$_i$ reflects results of a survey within one single OEM, *Impact*$_i$ comes from experts judgment rounds. The purpose of the scoring model is to support structured comparison and prioritization of transfer proposals rather than to provide universally valid absolute metrics. Taking the equation (1) and considering [69], error calculation for overall score $\sigma_{Score_{Estimated}}$ is derived:

$$\sigma_{score_{Est.}} = \sqrt{\sum_{1}^{i}\left[\left(\frac{l_i}{10}\cdot\sigma_{W_i}\right)^2 + \left(\frac{W_i}{10}\cdot\sigma_{l_i}\right)^2\right]} \quad (2)$$

Based on [70] we take $\sigma_{W_{\mathrm{i}}}$ as 7-10%. $\sigma_{I_{\mathrm{i}}}$ is the error margin for experts judgment, which cannot be easily quantified. There is no scientifically valid, universal percentage that indicate how much experts are wrong. The only indication is the agreement/deviation level of experts on single constraints. Due to that we treat our results as empirically derived indications based on which qualitative statements can be done, and recommend a quantitative treatment carefully.

6 CONCLUSION

This work presented a comparative analysis of aviation and automotive development practices along the V-model, focusing on autonomous driving systems and its functional safety. The objective was to identify transferable aviation practices that could further strengthen safety assurance for emerging, software-intensive automotive technologies. The analysis confirmed that a structured comparison is both feasible and meaningful, as both domains follow similar lifecycle models but differ significantly in process objectives, regulatory involvement, and assurance philosophy. Aviation has historically achieved higher operational safety through early and continuous authority involvement, disciplined system-level assurance, and rigorous governance throughout development. The most significant transfer potential was identified at the system level, where enhanced involvement of regulatory authorities and the use of structured assurance practices for common cause analysis (e.g., ARP4761-style approaches), and enhancing ASPICE assessments were found to substantially increase confidence in system safety. This influence naturally extends into software planning, reducing the need for additional isolated process measures. In the software requirements phase, aviation practices such as two-level requirement hierarchies, explicit conditioning and forcing of requirements attributes, and early prototyping were identified as highly relevant for managing the complexity of autonomous driving functions. As discussed in Section IV, the stricter requirements structuring, traceability mechanisms, and review practices commonly used in aviation directly address the increased system complexity and safety validation challenges introduced by autonomous driving functions. At the architecture and design level, both domains face increasing demands for high-performance computing driven by data-intensive and AI-based applications, motivating future comparison of centralization concepts. The coding phase represents the only lifecycle stage where automotive development is more prescriptive, due to explicit recommendations for coding standards such as MISRA, reflecting differing safety philosophies. For software implementation and integration, rigorous tool qualification, particularly following aviation-style DO-330 principles, was identified as a high-impact transfer candidate. Verification and independent validation were found to be difficult to compare directly; however, process-level analysis revealed comparable efficiency, while aviation's higher practical independence is largely achieved through authority involvement already addressed at the system level. Overall, the results demonstrate that selective transfer of aviation development practices, especially at system, governance, requirements, and tool qualification levels, offers a viable and effective path to further enhance the safety of autonomous driving systems.

RQ1 How can aviation and automotive be compared? **Answer** By systematically comparing normative guidance (ISO26262 versus DO-178C) and practical development in corresponding stages of the V-model in terms of prescriptiveness, rigor, expert judgment and safety record.

RQ2 What can be transferred from aviation to automotive? **Answer** System-level involvement approaches, planning rigor, requirements engineering methods, tool qualification rigor, independent verification were identified as promising candidates for a transfer.

RQ3 What is the quantified benefit of each transfer proposal? **Answer** We emphasize again that the quantification relies on empirical methods and reflects reality within the constraints and error margins described in the main text. Our results indicate that, except for the proposal requiring mandatory authority involvement, all proposed measures achieve a score above 5 and are therefore recommended. DO-330-style tool qualification received the highest rating, with a score of 9.42. Requirements-engineering practices fall within the

Journal of Autonomous Vehicles and Systems. Received February 07, 2026;
Accepted manuscript posted May 8, 2026. doi:10.1115/1.4071897

range of 6.15–6.69, while system-level proposals score between 4.54 and 6.38. Detailed results are provided in Table 3.

As closing remark we highlight that empirical weighting factors used in the feasibility quantification are based on survey responses collected within a single OEM. While the number of participants provides a meaningful initial dataset, the resulting weightings still reflect specific priorities and organizational context. Consequently, the numerical results should be interpreted as indicative rather than representative of the entire automotive industry. Future work involving multiple organizations and supply chain levels would support the validation and refinement of these weighting factors.

REFERENCES

[1] Adler, M. W., Peer, S., and Sinozic, T., 2019, "Autonomous, connected, electric shared vehicles (aces) and public finance: An explorative analysis," *Transportation Research Interdisciplinary Perspectives*.

[2] Hegde, R., Mishra, G., and Gurumurthy, K. S., 2011, "An insight into the hardware and software complexity of ecus in vehicles," In Advances in Computing and Information Technology, D. C. Wyld, M. Wozniak, N. Chaki, N. Meghanathan, and D. Nagamalai, eds., Vol. 198 of *Communications in Computer and Information Science*, Springer.

[3] Kochanthara, S., Dajsuren, Y., et al., 2022, "Painting the landscape of automotive software in github," In Proceedings of the 19th IEEE/ACM International Conference on Mining Software Repositories (MSR), ACM.

[4] Szalavetz, A., 2024, "In-house software development for software-defined vehicles: Major changes ahead in automotive value chains?," *International Journal of Automotive Technology and Management,* **24**(5), pp. 22–39.

[5] Uzzaman, A., et al., 2025, "Review on the safety and sustainability of autonomous vehicles: Challenges and future directions," *Control Systems and Optimization Letters,* **3**(1).

[6] Chialastri, A., 2012, "Automation in aviation," In *Automation*. InTech.

[7] Akkus, Y., and Annighofer, B., 2022, "Comparison of aviation and automotive standards and methods in terms of safety and cybersecurity," In Software Engineering 2022 Workshops, Gesellschaft für Informatik.

[8] Majidi, M., 2023, Number of transportation fatalities in the united states in 2021, by mode `https://www.statista.com/statistics/860124/us-transportation-fatalities-by-mode/` Accessed: 28 Oct. 2025.

[9] Savage, I., 2013, "Comparing the fatality risks in united states transportation across modes and over time," *The Economics of Transportation Safety,* **43**(1), pp. 9–22.

[10] Gerlach, M., Weißleder, S., and Hilbrich, R., 2011, Can cars fly? from avionics to automotive: Comparability of domain specific safety standards.

[11] Crots, K., Skentos, P., et al., 2014, A comparative analysis of aviation and ground vehicle software development standards.

[12] Schwierz, A., and Forsberg, H., 2018, "Design assurance evaluation of microcontrollers for safety critical avionics," In IEEE/AIAA Digital Avionics Systems Conference (DASC).

[13] Ledinot, E., Astruc, J.-M., et al., 2012, A cross-domain comparison of software development assurance standards HAL preprint.

[14] Baek, J., et al., 2023, "Adopting automotive software technology to aviation through a comparative analysis of software development standards," In Proceedings of the 42nd IEEE/AIAA Digital Avionics Systems Conference (DASC), pp. 1–4.

[15] Leiner, B., 2007, A comparison of partitioning operating systems for integrated systems `https://www.cs.unc.edu/~anderson/teach/comp790/papers/comparison.pdf`.

[16] Gaska, T., 2015, "Integrated modular avionics — past, present, and future," *IEEE Aerospace and Electronic Systems Magazine,* **30**(9), pp. 12–23.

[17] Gaska, T., 2017, "Model-based engineering for advanced integrated modular avionics: Focus and challenges," In Vertical Flight Society 73rd Annual Forum and Technology Display.

[18] Bandur, V., 2021, "Making the case for centralized automotive e/e architectures," *IEEE Transactions on Vehicular Technology,* **70**(2), pp. 1230–1245.

[19] Arcanjo, R. R., et al., 2023, "Verification and validation of embedded software in an automotive context: A systematic literature review," *Revista Científica Multidisciplinar Núcleo do Conhecimento*.

[20] Osborne, R. J., Hawkins, R. D., and McDermid, J., 2024, "Understanding safety engineering practice: Comparing safety as desired, required and observed," *Safety Science,* **172**.

[21] Darwesh, D. N., 2023, "Systematic validation and evaluation of a safety-critical middleware for integrated modular avionics," PhD thesis, University of Stuttgart.

[22] Kassab, M., 2018, "Testing practices of software in

Journal of Autonomous Vehicles and Systems. Received February 07, 2026;
Accepted manuscript posted May 8, 2026. doi:10.1115/1.4071897

safety-critical systems: Industrial survey," In Proceedings of the 13th International Conference on Software Technologies (ICSOFT), SciTePress.

[23] Franch, X., Seyff, N., Fricker, S., and Palomares, C., 2021, A study about the knowledge and use of requirements engineering standards in industry arXiv preprint.

[24] RTCA, 2011, Do-178c: Software considerations in airborne systems and equipment certification, Dec.

[25] ISO, 2018, Iso 26262: Road vehicles – functional safety.

[26] Rierson, L., 2013, *Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance* CRC Press.

[27] Rapita Systems, 2025, Efficient verification through the do-178c life cycle Document ID: MC-WP-011 DO-178C Verification v7.

[28] Bhaskaran, B., 2025, The importance of homologation in an evolving u.s. automotive market PDF Whitepaper, accessed 24 Nov. 2025.

[29] VDA QMC Working Group 13 / Automotive SIG, 2017, Automotive spice process assessment / reference model Accessed: 24 Nov. 2025.

[30] Automotive SPICE Certification, 2025, Automotive spice certification Accessed: 30 Dec. 2025.

[31] SAE International, 2010, Arp4754a: Guidelines for development of civil aircraft and systems, Dec.

[32] SAE International, 1996, Arp4761: Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment, Dec.

[33] Akkus, Y., et al., 2025, "Dependent failure analysis for automotive fail-operational mechatronic systems with aviation best-practices," In 2025 IEEE 25th International Symposium on Computational Intelligence and Informatics (CINTI).

[34] Azianti, I., 2014, "Iso 26262 automotive functional safety: Issues and challenges," *International Journal of Reliability and Applications,* **15**(2), pp. 151–161.

[35] ZVEI – Electro and Digital Industry Association, 2024, Software for safety-related automotive systems.

[36] Automotive SPICE, 2026, Process dimension.

[37] Krog, J., et al., 2022, "Towards a systems engineering methodology for architecture development of vehicle concepts," In Proceedings of NordDesign 2022.

[38] Granrath, R., et al., 2021, "A new systems engineering approach highlighting the need for a good abstraction level independent from decomposition," *Systems Engineering*.

[39] Krog, J., Akbas, C., Nolte, B., and Vietor, T., 2025, "Development of a functional and logical reference system architecture in automotive engineering," *Systems,* **13**(3), p. 141.

[40] Federal Aviation Administration, 2009, Requirements engineering management handbook DOT/FAA/AR-08/32.

[41] Chrysalidis, P., Hoeber, H., and Thielecke, F., 2023, "A semi-automated approach for requirement-based early validation of flight control platforms," *CEAS Aeronautical Journal,* **14**, pp. 271–280.

[42] Löbl, D., 2018, "Total capability approach for the development of safety-critical functions," PhD thesis, Technical University of Munich.

[43] Rana, R., 2014, "Early verification and validation according to iso 26262 by combining fault injection and mutation testing," *Communications in Computer and Information Science,* **457**.

[44] Abboush, M., Knieke, C., and Rausch, A., 2025, "Advancing real-time validation of automotive software systems via continuous integration and intelligent failure analysis," *Scientific Reports,* **15**, p. 32936.

[45] Wiecher, C., Mandel, C., Günther, M., Fischbach, J., Greenyer, J., Greinert, M., Wolff, C., Dumitrescu, R., Mendez, D., and Abel, A., 2023, Model-based analysis and specification of functional requirements and tests for complex automotive systems arXiv preprint.

[46] Janßen, M., Pfandzelter, T., Wang, M., and Bermbach, D., 2023, Supporting uavs with edge computing: A review of opportunities and challenges Tech. Rep. MCC.2023.3, TU Berlin, ECDF, Mobile Cloud Computing Research Group.

[47] Han, P., Hu, W., Zhai, Z., and Huang, M., 2024, "A model-based optimization method of arinc 653 multicore partition scheduling," *Aerospace,* **11**(11), p. 915.

[48] Rehm, F., et al., 2021, "The road towards predictable automotive high-performance platforms," In DATE 2021.

[49] Mauser, L., Zimmermann, E., Nedvědický, P., Eisenreich, T., Wäschle, M., and Wagner, S., 2025, Towards mixed-criticality software architectures for centralized hpc platforms in software-defined vehicles: A systematic literature review arXiv preprint.

[50] Lukic, B., Ahlbrecht, A., Friedrich, S., and Durak, U., 2023, "State-of-the-art technologies for integrated modular avionics and the way ahead," In 42nd IEEE/AIAA Digital Avionics Systems Conference (DASC), pp. 1–10.

[51] MISRA, 2013, Misra c:2012 – guidelines for the use of the c language in critical systems.

Journal of Autonomous Vehicles and Systems. Received February 07, 2026;
Accepted manuscript posted May 8, 2026. doi:10.1115/1.4071897

[52] Ashmore, R., Howe, A., Chilton, R., and Faily, S., 2022, "Programming language evaluation criteria for safety-critical software in the air domain," In Proceedings of the IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), pp. 230–237.

[53] Bagnara, R., Bagnara, A., and Hill, P. M., 2018, The misra c coding standard and its role in the development and analysis of safety- and security-critical embedded software arXiv preprint.

[54] RTCA, 2011, Do-330 – software tool qualification considerations.

[55] Conrad, M., Munier, P., and Rauch, F., 2011, Software tool qualification according to iso 26262.

[56] Slotosch, O., Wildmoser, M., Philipps, J., and Jeschull, R., 2012, Iso 26262 – tool chain analysis.

[57] Ibrahim, M., and Durak, U., 2021, Qualification considerations for simulations in avionics software engineering Lecture, Technical University Clausthal.

[58] Vogel, M., et al., 2020, "Metrics in automotive software development: A systematic literature review," *Journal of Software: Evolution and Process,* **32**(11).

[59] Souza, M., and de Carvalho, T., 2005, The fault avoidance and the fault tolerance approaches for increasing the reliability of aerospace and automotive systems Tech. Rep. SAE Technical Paper 2005-01-4157, SAE International.

[60] Akkus, Y., and Annighöfer, B., 2026, Forecast of the feasibility of aviation-to-automotive transfer of processes, methods, and tools by empirical quantification Technical paper, ongoing publication.

[61] De Marco, A., Mangano, G., and De Magistris, P., 2021, "Evaluation of project management practices in the automotive original equipment manufacturers," *Procedia Computer Science,* **181**, pp. 310–324.

[62] Hove, S. E., and Anda, B., 2005, "Experiences from conducting semi-structured interviews in empirical software engineering research," In Proceedings of the 11th IEEE International Software Metrics Symposium (METRICS'05), pp. 1–23.

[63] Akkus, Y., 2024, Cross-domain transfer of development methods, employee survey targeting automotive developers' preferences Survey period: 7 Oct. 2024 – 25 Nov. 2024.

[64] Runeson, P., Höst, M., Rainer, A., and Regnell, B., 2009, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering,* **14**, pp. 131–164.

[65] Lim, J., 2024, Building stronger teams: The impact of automotive surveys in 2024 Accessed: 6 Jan. 2025.

[66] Molléri, J. S., Petersen, K., and Mendes, E., 2020, "An empirically evaluated checklist for surveys in software engineering," *Information and Software Technology*.

[67] Proceffa, 2023, Mape: Guide to mean absolute percentage error Accessed: 15 Jan. 2025.

[68] Project Management Institute, 2013, *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Fifth Edition* Project Management Institute Accessed: 28 Oct. 2025.

[69] Taylor, 2001, Propagation of errors — basic rules Accessed: 31 Oct. 2025.

[70] Hunter, P., 2024, Margin of error and confidence levels: Make the most of your customer surveys Accessed: 13 Jun. 2025.