

Perspectives on Safety for Autonomous Vehicles

Rahul Razdan ^{1,*}, Raivo Sell ¹, M. Ilhan Akbas ²  and Mahesh Menase ³

¹ Department of Mechanical and Industrial Engineering, Tallinn University of Technology, 19086 Tallinn, Estonia

² Electrical Engineering and Computer Science Department, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA

³ Acclivis Technologies Pvt Ltd., Pune 411021, India

* Correspondence: rahul.razdan@taltech.ee

Abstract

Autonomy is enabled by the close connection of traditional mechanical systems with information technology. Historically, both communities have built norms for validation and verification (V&V), but with very different properties for safety and associated legal liability. Thus, combining the two in the context of autonomy has exposed unresolved challenges for V&V, and without a clear V&V structure, demonstrating safety is very difficult. Today, both traditional mechanical safety and information technology rely heavily on process-oriented mechanisms to demonstrate safety. In contrast, a third community, the semiconductor industry, has achieved remarkable success by inserting design artifacts which enable formally defined mathematical abstractions. These abstractions combined with associated software tooling (Electronics Design Automation) provide critical properties for scaling the V&V task, and effectively make an inductive argument for system correctness from well-defined component compositions. This article reviews the current methods in the mechanical and IT spaces, the current limitations of cyber-physical V&V, identifies open research questions, and proposes three directions for progress inspired by semiconductors: (i) guardian-based safety architectures, (ii) functional decompositions that preserve physical constraints, and (iii) abstraction mechanisms that enable scalable virtual testing. These perspectives highlight how principles from semiconductor V&V can inform a more rigorous and scalable safety framework for autonomous systems.

Keywords: autonomous; cyber-physical systems; governance; safety; scaling; validation; verification



Academic Editors: George Yanniss and Eva Michelaraki

Received: 9 October 2025

Revised: 6 November 2025

Accepted: 15 November 2025

Published: 18 November 2025

Citation: Razdan, R.; Sell, R.; Akbas, M.I.; Menase, M. Perspectives on Safety for Autonomous Vehicles. *Electronics* **2025**, *14*, 4500. <https://doi.org/10.3390/electronics14224500>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The rapid advancement of autonomous systems has brought about significant changes in various industries. From self-driving cars navigating urban streets to intelligent drones performing industrial inspections, these systems are beginning to assume significant roles in daily life [1]. Their growing presence, however, poses a major challenge: they must be trusted to operate safely in complex, unpredictable environments. This trust depends, in large part, on the ability to rigorously evaluate their behavior before deployment. In 2019, a broad industry report commissioned by the Society for Automotive Engineering (SAE) [2] outlined the significant open issues in validation and verification of these systems. An updated report in 2025 [3] acknowledged some progress but emphasized the large number of outstanding issues which remain unresolved. Critical conclusions from the reports included:

Cyber-physical Coupling: Autonomous systems represent a tight cyber-physical coupling between mechanical actuation, sensing, control theory, and high-dimensional information processing. Historically, the mechanical systems community has relied on deterministic models, bounded-uncertainty analysis, formal safety margins, and standards-driven certification frameworks to support validation and verification (V&V). In contrast, the information technology domain—particularly modern data-driven algorithms—operates with stochastic behavior, adaptive software artifacts, and non-deterministic failure modes, where correctness is often demonstrated empirically rather than proven. When these two assurance paradigms are composed within an autonomous system, their differing epistemic assumptions about evidence of safety, traceability of failure, and assignability of legal liability become exposed and unresolved. As a result, the V&V problem loses the separability that traditionally allowed mechanical systems to be certified independently of computational subsystems.

Current Assurance Protocols: Current safety assurance practice in both domains remains strongly process-centric (e.g., ISO 26262 [4]). However, these process frameworks do not directly account for the epistemic uncertainty introduced by machine learning-based perception, online adaptation, and environment-driven scenario explosion. Without a unified V&V framework capable of reasoning over hybrid discrete-continuous, deterministic-stochastic behaviors, the demonstration of safety for autonomous systems remains fundamentally unsolved.

There is a field where large-scale validation has succeeded: semiconductors. Driven by Moore's Law [5], semiconductor V&V has had a deep requirement for scalability and the result is quite different from either the mechanical system or information technology process oriented structures [6–8]. The fundamentals of semiconductor V&V consist of three very important concepts:

1. **Design Artifacts:** In semiconductors, design artifacts are inserted to enable the process of componentization and structured composition. This methodology effectively allows one to build/verify components and be assured they will still work when composed at a higher level.
2. **Integration Mathematics:** The integration of components is defined by a formal mathematics, and the combination builds a formal abstraction of reality which can be verified quickly and efficiently.
3. **Hierarchy of Abstractions:** Abstractions can scale on top of each other such that the higher level abstraction can represent a much larger population of potential targets.

Pulling everything together, the V&V of semiconductor system is effectively a mathematical induction proof where components are verified separately, and the various compositions of the components are verified at higher levels of abstraction (layout, circuits, logical graphs, microarchitectures, etc.). The result has been to successfully validate chips and systems with trillions of transistors.

In this perspectives article, we suggest that the techniques developed in the semiconductor space can potentially be useful in the cyber-physical space. When considering this point, two intuitive thoughts come to mind:

1. **Complexity:** The intuition is that cyber-physical systems are more complex and must deal with a more unpredictable environment. However, on all the fundamental points related to V&V, semiconductors have a direct analogy to cyber-physical systems. Semiconductors, like cyber-physical systems, fundamentally live in the world of physics (electromagnetics vs. Newtonian physics) and must comprehend safe ODDs. Testing semiconductors also involves thinking through complex scenarios. In fact, because semiconductors live in nano second timeframes, the number of executed scenarios is quite significant. One of the major differences between semiconductors

and cyber-physical systems is the relative scale where semiconductors must contend with billions of objects.

2. **System Safety:** The other intuition is that semiconductors are actually part of bigger cyber-physical systems. Thus, a safety protocol such as ISO 26262 must necessarily include semiconductors, and if so, what is the added value? In fact, when process oriented protocols such as ISO 26262 are applied to semiconductors, they focus on manufacturing faults and higher level system function. Critically, the deep mathematical processes which were employed to build the chip are abstracted away.

In this perspectives paper, the authors suggest that the techniques employed by semiconductors may be useful in the context of cyber-physical V&V, thus enabling far more robust arguments for safety. The outline of this paper has the following sections:

1. **V&V Traditional Methods:** This section takes a first-principle analysis of the existing V&V methods at the center of safety protocols in the mechanical and information technology space, and shows their evolution with early integration.
2. **Autonomy Safety Current Approaches:** This section discusses the open issues to the deepest forms of integration and introduction of artificial intelligence components.
3. **Semiconductor Inspired Techniques:** The last section introduces the use of semiconductor inspired techniques with three specific proposals to address the open issues in autonomy V&V.

Overall, the objective of this perspective article is to thoughtfully present the key ingredients from three communities (system safety, information technology validation, and semiconductor validation) which do not traditionally have much interaction, and suggest vectors of collaboration. These ideas are not presented as final answers, but as directions that may guide the development of more robust and resilient validation frameworks as autonomy continues to advance.

2. V&V Traditional Methods

Product liability in modern societies is grounded in the balance between consumer benefit and potential harm, where legal responsibility is determined by whether a product behaves in line with reasonable expectations and by the severity of harm caused when those expectations are violated. For safety-critical autonomous systems, these expectations are still evolving, which complicates how liability, assurance, and acceptable risk are defined. The overarching governance framework—formed by laws, regulations, and judicial precedent—shapes both system specifications and the validation (ensuring the design meets intended use) and verification (ensuring the system is built correctly) processes. The Master Verification and Validation (MaVV) framework operationalizes “reasonable due diligence” through defining the Operational Design Domain (ODD), measuring test coverage, and incorporating post-deployment field response. In practice, MaVV is implemented via Minor V&V workflows that generate, execute, and evaluate tests across physical and virtual environments, balancing scalability with realism. Because exhaustive testing is infeasible, MaVV seeks to build a defensible, probabilistic argument that the system has been sufficiently tested for its intended operational domain, while continuously improving through field feedback. More detail on this process can be found in [Appendix A](#).

In practice, Verification and Validation (V&V) processes face the fundamental challenge of addressing extremely large Operational Design Domain (ODD) spaces while operating under constraints of limited computational and experimental capacity, high evaluation cost, and strict time-to-market pressures. Achieving sufficient coverage within these constraints requires careful prioritization of test effort and intelligent allocation of validation resources. To manage this complexity, V&V methodologies are traditionally organized into two overarching categories: **Physics-Based** and **Decision-Based** approaches. Each of these

paradigms embodies distinct assumptions, tools, and validation logics for demonstrating product correctness and safety within its intended operational domain. The following sections discuss the defining characteristics and comparative advantages of each approach.

2.1. Traditional Physics-Based Execution

Within the Master Verification and Validation (MaVV) framework, two critical dimensions determine assurance effectiveness: the efficiency of the Minor V&V (MiVV) engine—that is, the capability to generate, execute, and evaluate tests at scale—and the strength of the completeness argument, which justifies that validation has been sufficient relative to the system’s potential to cause harm. Historically, complex mechanical or electro-mechanical systems such as automobiles, aircraft, and industrial machinery demanded highly developed V&V processes. These legacy systems exemplify a broader class of products governed by a Physics-Based Execution (PBE) paradigm.

In the PBE paradigm, the execution model of the system—whether in simulation or physical operation—is grounded in the laws of physics, characterized by continuity and monotonicity. Small perturbations in system inputs or environmental conditions typically produce proportionally small and predictable changes in outcomes. This inherent regularity allows for deterministic reasoning, structured test-space reduction, and relatively stable causal relationships between design parameters and observed performance. Consequently, traditional V&V techniques in PBE systems could rely on physical testing, analytical modeling, and controlled experimentation to build confidence in system correctness and safety.

This key insight has enormous implications for V&V because it greatly constrains the potential state-space to be explored. Examples of this reduction in state-space include:

- (1) Scenario Generation: One need only worry about the state space constrained by the laws of physics. Thus, objects which defy gravity cannot exist. Every actor is explicitly constrained by the laws of physics.
- (2) Monotonicity: In many interesting dimensions, there are strong properties of monotonicity. As an example, if one is considering stopping distance for braking, there is a critical speed above which there will be an accident. Critically, all the speed bins below this critical speed are safe and do not have to be explored.

Mechanically, in traditional PBE fields, the philosophy of safety regulation (ISO 26262 [4], AS9100 [9], etc.) builds the safety framework as a process, where:

1. Failure mechanisms are identified;
2. A test and safety argument is built to address the failure mechanism;
3. There is an active process by a regulator (or documentation for self-regulation) which evaluates these two, and acts as a judge to approve/decline.

Traditionally, faults considered were primarily mechanical failure. Appendix B provides an example implementation of the process for illustration.

Finally, the regulations have a strong idea of safety levels with Automotive Safety Integrity Levels (ASIL). Airborne systems follow a similar trajectory (pun intended) with the concept of Design Assurance Levels (DALs). A key part of the V&V task is to meet the standards required at each ASIL level.

Historically, a sophisticated suite of Verification and Validation (V&V) techniques has been developed to ensure the safety and reliability of traditional automotive systems. These methods have relied heavily on structured physical testing regimes, frequently validated or certified by regulatory bodies and independent testing organizations such as TÜV SÜD [10]. Over time, the scope of these activities has expanded through the integration of virtual physics-based models, which augment physical testing by enabling simulation-driven analysis of complex design tasks—ranging from vehicle body aerodynamics [11] to tire performance and road–tire interaction modeling [12].

The general structure of such physics-based modeling frameworks involves a sequential process of model generation, predictive execution, and model correction. In this flow, simulation models are constructed to replicate the relevant physical behaviors of the system, executed to predict performance across a broad range of operating conditions, and iteratively refined based on empirical validation data. This predictive capacity allows engineers to explore larger portions of the Operational Design Domain (ODD) efficiently and systematically. However, because these simulations are constrained by the computational complexity of faithfully representing physical phenomena, virtual physics-based simulators often exhibit limited performance and may require specialized hardware acceleration to achieve feasible run times for large-scale validation studies.

In summary, the key underpinnings of the PBE paradigm from a V&V point of view are:

1. Constrained and well-behaved space for scenario test generation.
2. Expensive physics based simulations.
3. Regulations focused on mechanical failure.
4. In safety situations, regulations focused on a process to demonstrate safety with a key idea of design assurance levels.

2.2. Traditional Decision-Based Execution

In their foundational form, Information Technology (IT) systems were not considered safety-critical, and as a result, comparable levels of legal liability have not historically attached to IT products. Nonetheless, the rapid expansion and pervasive integration of IT systems into global economic infrastructure mean that failures in large-scale consumer platforms can now have catastrophic economic and societal consequences [13]. Consequently, robust Verification and Validation (V&V) practices remain essential to maintaining system reliability, security, and trust.

While IT systems follow the same overarching V&V framework described earlier, they diverge significantly from physics-based systems along two key dimensions: the execution paradigm and the nature of error sources. Unlike the Physics-Based Execution (PBE) paradigm, IT systems operate under a Decision-Based Execution (DBE) model. In the DBE paradigm, there are no intrinsic physical constraints governing functional behavior—no continuity, monotonicity, or conservation principles that naturally limit the system's state space. As a result, the Operational Design Domain (ODD) for IT systems is vast and combinatorially complex, making comprehensive test generation and coverage demonstration exceedingly difficult.

To address these challenges, the IT community has developed a series of structured processes and formalized methodologies aimed at improving the robustness and repeatability of V&V. These processes incorporate systematic test planning, layered validation environments, formal verification for critical modules, and continuous integration pipelines to identify and mitigate defects throughout the development lifecycle. These include:

- (1) Code Coverage: Here, the structural specification of the virtual model is used as a constraint to help drive the test generation process. This is carried out with software or hardware (Register Transfer Level (RTL) code).
- (2) Structured Testing: A process of component, subsection, and integration testing has been developed to minimize propagation of errors.
- (3) Design Reviews: Structured design reviews with specs and core are considered best practice.

A representative framework exemplifying structured process maturity in software development is the Capability Maturity Model Integration (CMMI), developed at Carnegie Mellon University [14]. CMMI defines a hierarchical set of processes, metrics, and continuous-improvement practices designed to ensure systematic delivery of high-quality

software. While originally conceived for conventional software engineering, substantial components of the CMMI architecture can be adapted for Artificial Intelligence (AI) systems—particularly in cases where AI modules are integrated as replacements or enhancements for traditional software components. Such adaptation provides a governance-oriented approach to managing quality, consistency, and traceability across increasingly complex digital systems.

Testing within the Decision-Based Execution (DBE) domain can be philosophically decomposed into three categories of epistemic uncertainty:

1. Known knowns—Issues that are both identified and understood, representing defects detectable through standard debugging or verification processes.
2. Known unknowns—Anticipated risks or potential failure modes whose specific manifestations or root causes remain uncertain but are targeted through exploratory or stress testing.
3. Unknown unknowns—Completely unanticipated issues that emerge without prior awareness, often revealing fundamental gaps in system design, understanding, or testing scope.

Among these, the unknown unknowns category poses the greatest challenge to DBE-oriented V&V, as it represents failures that elude conventional test design. To address this problem, pseudo-random test generation and related stochastic exploration methods have become key techniques for uncovering emergent faults in complex software and AI systems [15]. These methods enable broader, less biased coverage of the massive ODD space inherent to decision-based systems, thereby enhancing the resilience and completeness of the overall verification strategy.

In summary, the key underpinnings of the DBE paradigm from a V&V point of view are:

- (1) Unconstrained and not well-behaved execution space for scenario test generation.
- (2) Generally, less expensive simulation execution (no physical laws to simulate).
- (3) V&V focused on logical errors not mechanical failure.
- (4) Generally, no defined regulatory process for safety critical applications. Most software is “best efforts”.
- (5) “Unknown-unknowns” a key focus of validation.

A **key implication of the DBE space** is that the idea from the PBE world of building a list of faults and building a safety argument for them is antithetical to the focus of DBE validation.

2.3. Mixed Domain Architectures

The fundamental characteristics of Digital-Based Engineered (DBE) systems pose significant challenges when applied to safety-critical domains. Nevertheless, the Information Technology (IT) sector has been one of the defining megatrends of the past fifty years, driving profound transformation across global industries. Appendix C provides more details on the driving economic and technical forces.

In doing so, it has built vast ecosystems encompassing semiconductors, operating systems, communications, and application software. Today, leveraging these ecosystems is essential for the success of nearly all modern products, making mixed-domain safety-critical systems—where DBE and traditional Physics-Based Engineering (PBE) components coexist—an unavoidable reality.

Mixed-domain architectures can generally be categorized into three paradigms, each presenting distinct Verification and Validation (V&V) requirements:

1. Mechanical Replacement—characterized by a large PBE component and a small DBE component.

2. Electronic Adjacent—where PBE and DBE systems operate as separate but interacting domains.
3. Autonomy—dominated by the DBE component, with the PBE portion playing a supporting role.

A representative example of the mechanical replacement paradigm is Drive-by-Wire functionality, where traditional mechanical linkages are replaced by electronic components (hardware and software). In early implementations, these hybrid electronic-mechanical systems were physically partitioned as independent subsystems, allowing the V&V process to resemble traditional mechanical verification workflows. Over time, regulatory frameworks evolved to recognize electronic failure modes, as reflected in emerging standards such as SOTIF (Safety of the Intended Functionality) (see Table 1) [16].

Table 1. Differences between SOTIF and ISO 26262.

Aspect	ISO 26262	SOTIF
Focus	System faults and malfunctions	Hazards due to functional insufficiencies
Applicability	All safety-critical systems	Primarily ADAS and autonomous systems
Hazard Source	Hardware and software failure	Limitations in functionality, unknown scenarios
Methods	Fault avoidance and control	Scenario-based testing

The paradigm of maintaining separate physical subsystems offers clear advantages in terms of V&V simplicity and inherent safety isolation. However, it also introduces substantial disadvantages, notably component skew, increased material costs, and system integration inefficiencies. As a result, a major industry trend has been to consolidate hardware into shared computational fabrics interconnected through high-reliability networks, while maintaining logical or virtual separation between safety-critical and non-safety functions.

From a V&V standpoint, this architectural shift transfers much of the assurance burden from the physical to the virtual domain. The virtual backbone—including elements such as the Real-Time Operating System (RTOS), hypervisors, and inter-process communication mechanisms—must now be verified to a very high standard. Ensuring that this backbone preserves determinism, isolation, and predictable timing behavior becomes central to maintaining safety assurance in these mixed-domain systems.

Infotainment systems are a prime example of the Electronics-Adjacent integration paradigm. In these architectures, an independent IT infrastructure operates alongside the safety-critical control infrastructure, allowing each domain to be validated separately from a V&V standpoint. However, infotainment platforms inherently introduce powerful communication technologies—such as 5G, Bluetooth, and Wi-Fi—that expose the cyber-physical system to potential external interference.

From a safety perspective, the most straightforward method of maintaining isolation would be physical separation between IT and safety-critical systems. In practice, however, this is rarely implemented, as connectivity is essential for Over-the-Air (OTA) software updates and in-field maintenance. OTA capabilities deliver enormous economic and operational advantages, reducing lifecycle maintenance costs and enabling rapid deployment of patches and new features. Consequently, V&V activities must extend beyond functional correctness to include verification that virtual safeguards—such as firewalls, trusted execution environments, and secure boot mechanisms—are robust against malicious intrusion or unintended cross-domain influence.

The third and most complex integration paradigm arises in the context of autonomy. Here, the DBE subsystems dominate: processes such as sensing, perception, localization, and path planning encapsulate and control the traditional mechanical PBE functionality.

As illustrated conceptually in Figure 1, the autonomous execution paradigm can be viewed as consisting of four hierarchical layers:

- Layer 4 (Core Physics): the physical world governed by deterministic, well-understood PBE properties.
- Layer 3 (Actuation and Edge Sensing): traditional control and sensor interfaces that largely retain PBE characteristics.
- Layer 2 (Computation and AI): software-intensive DBE components responsible for inference, prediction, and decision-making.
- Layer 1 (Design-for-Experiment V&V Layer): the outermost assurance framework, which must validate a system with inherently PBE behaviors through an interface dominated by stochastic, data-driven DBE-AI processes.

This multi-layered structure presents an unprecedented challenge for V&V. It requires reconciling the determinism and physical certainties of PBE with the probabilistic, adaptive, and data-dependent behaviors characteristic of DBE-AI systems.

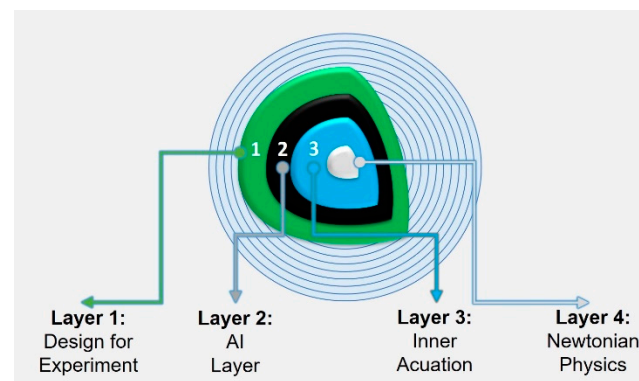


Figure 1. Conceptual Layers in Cyber-Physical Systems.

3. Autonomy V&V Current Approaches

For safety-critical systems, the evolution of Verification and Validation (V&V) practices has been tightly coupled to the development of regulatory standards frameworks such as ISO 26262 in the automotive domain. Key elements of this framework include:

1. System Design Process—A structured development-assurance approach for complex systems, integrating safety certification within the broader system-engineering process.
2. Formalization—The explicit definition of system operating conditions, intended functionalities, expected behaviors, and associated risks or hazards requiring mitigation.
3. Lifecycle Management—The management of components, systems, and development processes across the entire product lifecycle, from concept through decommissioning.

The primary objective of these frameworks is to ensure that systems are meticulously and formally specified, their expected behaviors and potential failure modes are anticipated and mitigated, and their lifecycle impacts are comprehensively understood.

With the emergence of software-driven architectures, safety-critical V&V methodologies evolved to preserve the rigor of the original system design approach while incorporating software as an integral system component. In this context, software elements were assimilated into existing structures for fault analysis, hazard identification, and lifecycle management. However, the introduction of software required fundamental extensions to traditional V&V concepts.

For example, in the airborne systems domain, the DO-178C standard [17] (“Software Considerations in Airborne Systems and Equipment Certification”) redefined the notion of

hazards—from physical failure mechanisms to functional defects—recognizing that software does not degrade through physical wear. Similarly, lifecycle management practices were updated to reflect modern software-engineering workflows, including iterative development and configuration control. The introduction of Design Assurance Levels (DALs) further enabled the integration of software components into system design, functional allocation, performance specification, and the V&V process. These mechanisms parallel the role of SOTIF in the automotive sector, providing a means to align software functionality with system-level safety objectives.

Moving beyond conventional software, Artificial Intelligence (AI) introduces a fundamentally different paradigm—one based on learning rather than explicit programming. In this learning paradigm, the system undergoes a training phase in which it derives internal representations and decision rules directly from data. Learning, in this context, is implemented through optimization algorithms that iteratively minimize a defined error or loss function, effectively creating a data-driven form of software development.

However, as Table 2 illustrates, there are profound structural and operational differences between AI-based software and traditional deterministic software. In conventional systems, behavior is fully specified and traceable to explicit code paths. In contrast, AI systems depend on large datasets, statistical generalization, and model architectures whose internal logic may be opaque, adaptive, and context-dependent.

Table 2. Contrast of conventional and machine learning algorithms.

Conventional Algorithm	ML Algorithms	Comment
Logical Theory	No Theory	In conventional algorithms, one needs a theory of operation to implement the solution. ML algorithms can often “work” without a clear understanding of exactly why they work.
Analyzable	Not Analyzable	Conventional algorithms are encoded in a way one can see and analyze the software code. Most validation and verification methodologies rely on this ability to find errors. ML algorithms offer no such ability, and this leaves a large gap in validation.
Causal	Correlation	Conventional algorithms have built in causality and ML algorithms discover correlations. The difference is important if one wants to reason at a higher level.
Deterministic	Non-Deterministic	Conventional algorithms are deterministic in nature, and ML algorithms are fundamentally probabilistic in nature.
Known Computational Complexity	Unknown Computational Complexity	Given the analyzable nature of conventional algorithms, one can build a model for computational complexity. That is, how long will it take the algorithm to work. For ML techniques, no generic method exists to evaluate computational complexity.

These distinctions give rise to three central challenges—the so-called “elephants in the room” for AI assurance:

1. **AI Component Validation**—How can we establish verification confidence for components whose internal behavior is learned rather than explicitly defined [18,19]?

2. AI Specification—How do we formally specify and bound the expected behavior of a system whose logic emerges from data rather than design [20]?
3. Intelligent Scaling—How do we ensure system reliability, safety, and explainability as AI systems scale in complexity, data volume, and autonomy?

These challenges redefine the boundaries of V&V, demanding new methods that integrate statistical assurance, behavioral monitoring, and explainability into the traditional systems engineering framework.

3.1. AI Component Validation

Both the automotive and airborne sectors have responded to the emergence of AI by treating it as a form of “specialized software” within evolving standards frameworks such as ISO 8800 [17,21]. This strategy has clear advantages: it enables reuse of decades of foundational work in mechanical safety assurance and software verification, thereby maintaining continuity with well-established engineering and certification practices.

However, this approach introduces a fundamental challenge—AI systems do not rely on traditional hand-written code, but instead produce behavior through data-driven model training. In effect, AI development substitutes data-generated logic for explicitly programmed logic. This distinction creates profound implications for V&V.

In conventional software engineering, key assurance mechanisms—such as coverage analysis, code review, and version control—operate on deterministic artifacts. In AI-enabled systems, these same assurance activities must be re-interpreted and extended to address the unique properties of learned models, training data, and the associated development pipelines (see Table 3). The result is a need to bridge legacy software assurance methodology with emerging practices for data governance, model validation, and continual performance monitoring.

Table 3. Software vs AI/ML Validation Methods.

V&V Technique	Software	AI/ML
Coverage Analysis:	Code Structure provides basis of coverage	No structure
Code Reviews:	Crowd source expert knowledge	No Code to Review
Version Control	Careful construction/release	Very Difficult with data

These differences introduce profound challenges for intelligent test generation and for making any defensible argument of completeness. This remains an area of active and rapidly evolving research, with two primary investigative threads emerging:

1. Training Set Validation—Because the learned AI component is often too complex for direct formal analysis, one strategy is to examine the training dataset in relation to the Operational Design Domain (ODD). The objective is to identify “gaps” or boundary conditions where the data distribution may not adequately represent real-world scenarios, and to design targeted tests that expose these weaknesses [22].
2. Robustness to Noise—Another approach, supported by both simulation and formal methods [23], is to assert higher-level system properties and test the model’s adherence to them. For instance, in object recognition, one might assert that an object should be correctly classified regardless of its orientation or lighting conditions.

Overall, developing robust, repeatable, and certifiable methods for AI component validation [18,19] remains a critical and unsolved research challenge, particularly for “fixed-function” AI components—that is, models whose parameters remain constant under version control. Yet, many emerging AI applications employ adaptive or continuously learning models, where the AI component evolves dynamically over time. Establishing

validation frameworks for these morphing systems represents one of the most significant frontiers for future research in AI assurance.

3.2. AI Specification

For well-defined systems with clear system-level abstractions, AI and ML components substantially increase the difficulty of intelligent test generation. When a golden specification exists, it is possible to follow a structured validation process and even constrain or “gate” AI outputs using conventional safeguards. However, some of the most compelling applications of AI arise precisely in domains where such specifications are ill-defined or infeasible to articulate through conventional programming.

In these Specification-Less/Machine Learning (SLML) contexts, not only is it challenging to create representative or “interesting” tests, but evaluating the correctness of the resulting behaviors becomes even more problematic. Notably, many critical subsystems in autonomous vehicles—such as perception, localization, and path planning—fall squarely into this SLML category. To date, two principal strategies have emerged to address the lack of explicit specifications: **Anti-Spec** and **AI-Driver**.

In the absence of a positive formal specification, one approach is to define correctness by negation—that is, through anti-specifications. The simplest anti-spec, for example, is the principle of *avoiding accidents*. Building on early work by Intel, IEEE 2846, “Assumptions for Models in Safety-Related Automated Vehicle Behavior” [24], formalizes this concept by establishing a framework for defining the minimum set of assumptions about the reasonably foreseeable behaviors of other road users. For each scenario, it specifies assumptions about kinematic parameters—such as speed, acceleration, and possible maneuvers—to model safe system responses.

Despite its utility, this approach presents three challenges:

1. Making a convincing completeness argument for all foreseeable scenarios.
2. Developing machinery for conformance checking against the standard.
3. Connecting these assumptions to a liability and governance framework [25].

Whereas IEEE 2846 adopts a bottom-up, technology-driven perspective, Koopman and Widen [26] have proposed a top-down framework—the concept of an AI Driver—which seeks to replicate the competencies of a human driver in a complex real-world environment.

Key elements of the AI-Driver concept include:

1. Full Driving Capability—The AI Driver must handle the entire driving task, encompassing perception (environment sensing), decision-making (planning and response), and control (executing maneuvers such as steering or braking), including social norms and unanticipated events.
2. Safety Assurance—AVs must be subject to rigorous safety processes similar to those used in aviation, including failure analysis, risk management, and fallback safety.
3. Human Equivalence—The AI Driver should meet or exceed the performance of a competent human driver, obeying traffic laws, handling rare “edge cases,” and maintaining continuous situational awareness.
4. Ethical and Legal Responsibility—The system must operate within moral and legal frameworks, including handling ethically charged scenarios and questions of liability.
5. Testing and Validation—Robust testing, simulation, and real-world trials are essential to validate performance across diverse driving conditions, including long-tail and edge-case scenarios.

Although conceptually powerful, the AI-Driver approach faces significant challenges. First, the notion of a “reasonable driver” is not even precisely defined for humans; it is the result of decades of legal and social distillation. Second, encoding this level of competence into a formal standard would entail immense complexity and regulatory

latency—it may take many years of legal evolution to converge on a broadly accepted definition of “AI reasonableness.”

Current State of Specification Practice

Presently, the state of AI specification remains immature across both Advanced Driver-Assistance Systems (ADAS) and Autonomous Vehicles (AVs). Despite widespread deployment, ADAS products exhibit substantial behavioral divergence and lack consistent definitions of capability or completeness. When consumers purchase an ADAS-equipped vehicle, it is often unclear what level of autonomy or driver assistance they are actually obtaining.

Independent testing organizations such as AAA, Consumer Reports, and the Insurance Institute for Highway Safety (IIHS) have repeatedly documented these shortcomings [27]. In 2024, the IIHS introduced a new ratings program to evaluate safeguards in partially automated driving systems; out of 14 systems tested, only one achieved an “acceptable” rating, underscoring the urgent need for stronger measures to prevent misuse and ensure sustained driver engagement [28]. As of today, the only non-process-oriented regulation currently active in this domain is the NHTSA rulemaking around Automatic Emergency Braking (AEB) [29].

These results highlight a central truth: until the industry develops credible frameworks for specifying and evaluating AI behavior—whether through Anti-Spec, AI-Driver, or future hybrids—V&V for autonomy will remain incomplete by design.

3.3. Intelligent Test Generation

Recognizing the importance of intelligent scenario generation for testing, three principal styles of intelligent test generation are currently active in the safety-critical AI domain: physical testing, real-world seeding, and virtual testing.

3.3.1. Physical Testing

Physical testing remains the most expensive and resource-intensive method for validating functionality, yet it provides the highest degree of environmental fidelity. A notable exemplar is Tesla, which has transformed its production fleet into a large-scale distributed testbed. Within this framework, Tesla employs a sophisticated data pipeline and deep learning infrastructure to process massive volumes of sensor data [30]. Each scenario encountered on the road effectively becomes a test case, with the driver’s corrective action serving as the implicit criterion for correctness.

Behind the scenes, this process—often referred to as the Massive Validation and Verification (MaVV) flow—is supported by extensive databases and specialized supercomputers, such as Tesla’s Dojo [31]. This methodology ensures that all test scenarios are, by definition, real and contextually valid. However, the approach faces several limitations:

1. The rate of novel scenario discovery in real-world driving is inherently slow.
2. The geographical and demographic bias of Tesla’s fleet restricts coverage to markets where the vehicles are deployed.
3. The process of data capture, error identification, and corrective model update is complex and time-intensive—analogous to debugging computers by analyzing crash logs after failure.

3.3.2. Real-World Seeding

A complementary strategy is real-world seeding, in which physical observations are used as seeds for more scalable virtual testing. The Pegasus project in Germany pioneered this approach, emphasizing scenario-based testing that starts from empirically observed data and expands into systematic virtual variations [32].

Similarly, researchers at Warwick University have advanced methods for test environment modeling, safety analysis, and AI-driven scenario generation. One of their key contributions is the Safety Pool Scenario Database [33], which curates challenging and rare test cases. While these databases offer valuable coverage of “interesting” situations, their completeness remains uncertain, and they are prone to over-optimization when AI systems inadvertently learn to perform well only on the available test set.

3.3.3. Virtual Testing

The third and increasingly dominant approach is virtual testing, enabled by high-fidelity simulation and symbolic scenario generation. A landmark development in this area is ASAM OpenSCENARIO 2.0 [34], a domain-specific language (DSL) designed to enhance the development, testing, and validation of Advanced Driver-Assistance Systems (ADAS) and Automated Driving Systems (ADS). This language supports hierarchical scenario composition, allowing for symbolic definitions that can scale in complexity. Beneath the symbolic layer, pseudo-random test generation enables large-scale scenario creation, which helps expose previously unseen “unknown-unknowns.”

Beyond component validation, efforts such as UL 4600, “Standard for Safety for the Evaluation of Autonomous Products” [35], extend safety assurance across the full product lifecycle. Like ISO 26262 and SOTIF, UL 4600 employs a structured safety-case methodology, requiring developers to document and justify how autonomous systems meet safety goals. The standard explicitly emphasizes scenario completeness, edge-case validation, and human–machine interaction. However, UL 4600 remains primarily a process standard—it guides how to argue for safety but offers limited prescriptive direction on how to resolve the foundational challenges of AI validation, specification, or scaling.

3.3.4. Process-Centric Limitations

Indeed, nearly all existing standards and regulatory mechanisms remain process-centric. They rely on the developer’s safety case—either self-certified or regulator-approved—to establish confidence. This paradigm has a critical weakness: developers lack practical methods to overcome the core technical limitations, while regulators lack objective tools to assess the completeness or validity of those claims.

Although these methodologies have advanced the state of the art, a fundamental issue remains: how can testing—physical or virtual—be scaled to adequately explore the Operational Design Domain (ODD)?

Moreover, within virtual environments, the question of model abstraction arises. Should simulations employ highly abstract behavioral models to enable large-scale statistical exploration, or detailed physics-based models for precision and realism? The answer typically depends on the verification objective, but there is currently no standardized method to link results across abstraction levels.

A critical missing capability is the ability to decompose the verification problem into manageable, modular components and then recompose the results into a coherent assurance argument. Such compositional verification techniques are mature in semiconductor design, where they underpin the success of large-scale digital systems verification, but have yet to be effectively adapted to cyber-physical and AI-based systems.

4. Semiconductor V&V as Inspiration for Cyber-Physical Research

This section proposes three significant research vectors, largely inspired by semiconductors, which have the potential to move the field forward. These are:

- (1) Guardian Accelerant Model.
- (2) Functional Decomposition.

(3) Pseudo Physical Scaling Abstraction.

4.1. Guardian Accelerant Model

In the field of computer architecture, automated learning methods are quite common in the form of performance accelerators such as branch or data prediction. Prediction algorithms, based on machine learning/AI techniques, are “trained” by the ongoing execution of software and predict to very high accuracy the branching or data values critical to overall performance execution. However, since this is only a probabilistic guess, a vast well defined “guardian” machinery is built which detects errors and unwinds the erroneous decisions. The combination leads to enormous acceleration in performance with safety.

Traditionally, the critical elements of autonomous cyber-physical systems are based on the AI training and inference paradigm. As the complexity and safety considerations have grown, a non-AI based safety layer has been growing. In fact, one of the more interesting systems consists of independent risk assessment guardians [36] which are running parallel to the core algorithm. Today, these techniques are somewhat ad hoc and often in response to “patching” the current bug in the core AI algorithm. An interesting line of research would be to formalize the AV and Guardian framework. In this framework, the Guardian, which should be specified more formally, would set the bounds within which the AV algorithm would operate. This decomposition has some very interesting properties:

- (1) Training Set Bounding: The core power of AI is to predict reasonable approximations between training points with high probability. However, there is also the idea that the training set is not complete relative to the current situation. In this decomposition, the AI algorithms can continue to be optimized for best guess, but the Guardian can be configured for the bounding box of expectations of the AI.
- (2) Validation and Verification: Given a paradigm for a Guardian and well-established rules for interaction between the Guardian and the AV/AI, a large part of the safety focus moves to the Guardian, a somewhat simpler problem. The V&V for the AV moves a very hard but non-safety critical problem of performance validation.
- (3) Regulation: A very natural role for regulation and standards would be specify the bounds for the Guardian while leaving the performance optimization to industry.

4.2. Functional Decomposition

As shown in Figure 1, the cyber-physical problem is stymied by a layer of DBE processing in a world which is fundamentally PBE. This problem is somewhat akin to the problems caused by numerical approximation [37]. That is, the underlying mathematical properties of continuity and monotonicity are broken by rounding, truncation, and digitization. With a deeper understanding of the underlying functions, numerical mathematics has developed techniques to deal with the filter of digitization.

Cyber-Physical V&V must build a similar model where the underlying properties of the PBE world are preserved through the DBE layer. This is a rich area of research and can take many forms. These include:

- (1) Invariants: The PBE word implies invariants such as real-world objects can only move so fast and cannot float or disappear or that important objects (cars) can be successfully perceived in any orientation. The invariants can be part of a broader anti-spec and basis of a validation methodology.
- (2) PBE World Model: A standard for describing the static and dynamic aspects of a PBE world model are interesting. If such a standard existed, both the active actors as well as infrastructure could contribute to building it. In this universal world model, any of the actors could communicate safety hazards to all the players through a V2X communication paradigm. Note, a universal world model (annotated by source)

becomes a very good risk predictor when compared to the world model built from the host cyber-physical system.

- (3) Intelligent Test Generation: With a focus on the underlying PBE, test generation can focus on transformations to the PBE state graph and the task of the PBE/DBE differences can be handled by other mechanisms such as described in #2.

4.3. Pseudo Physical Scaling Abstractions

In semiconductors, the use of virtual models has developed to a point where multi-billion-dollar business models are buttressed by a combination of virtual models, associated processes for characterization, and standards-based signoff methodologies. The result is a design and validation flow which has successfully built enormously complex multibillion component semiconductors. The key processes which enable this flow are:

- (1) Design Constraints: Semiconductors insert design artifacts which enable the properties of separation-of-concerns and abstraction. These design artifacts are often at the cost of performance, power, and product cost.
- (2) Mathematics: The constrained design now can use higher level more abstract mathematical forms which have better properties of scalability for both design and verification.
- (3) Abstraction and Synthesis: The higher level of mathematics is supported by processes of abstraction, building the mathematical form from a lower level, and synthesis, building the lower form from the higher abstraction.
- (4) Global Abstraction Enforcement: A robust machinery validates the key aspects which enable the higher level abstraction.

Finally, the design constraints, the new mathematics, and other required checks are mechanically managed by a multi-billion dollar Electronic Design Automation (EDA) software industry.

As an example, the lowest level in semiconductors is the world of physics with Maxwell's equations being the governing paradigm. However, unlike cyber physical systems today, semiconductors insert the design constraints of the separation between components (transistors, capacitor, etc.) and interconnect. This separation-of-concerns allows the governing mathematics to shift from Maxwell's equations to the solution of Partial Differential Equations (PDEs) in traditional circuit theory, a simpler mathematics [38]. Algorithms, implemented by software, exist to abstract the physical layer to the electrical layer and synthesize from the circuit graph to the physical layout. Finally, there are many tools which operate on the physical design and the circuit graph to enforce electrical isolation. The consequences of this shift are profound:

- (1) Maxwell Equation level Validation can be limited at the component level.
- (2) Electrical V&V can happen at the circuit graph level.
- (3) By design and through various tool verification, isolation properties are enforced.

Finally, all the abstraction levels and most of the meta-data are specified in open standards around which the EDA industry builds products [39].

The circuit graph is just the first step. With semiconductors this process is recursively performed with higher and higher levels of abstraction with similar properties. Figure 2 demonstrates this process for the initial levels of the semiconductor design process. The next interesting stage after the circuit graph simplification is to have the components be only unidirectional digital components, and this allows one to use the mathematics of Discrete Logic. As an aside, this was the Master's thesis for Claude Shannon [40]. In the time domain, one can introduce the design artifacts of latches/flip-flops and clocking to introduce the synchronous logic abstraction. This allows one to use the mathematics of the Finite State Machines. This process continues to go higher into mathematical abstractions such as computer architecture, operating systems, and higher-level software layers.

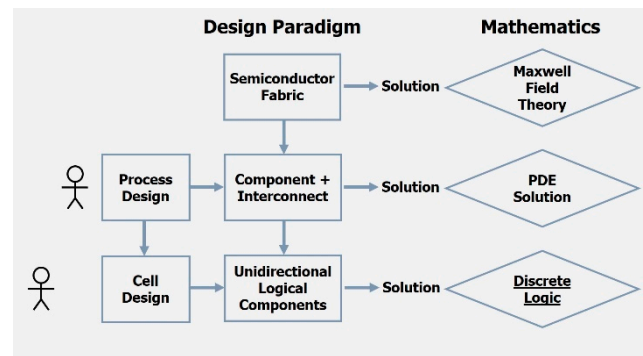


Figure 2. Semiconductor Design Artifacts, Abstractions, and Mathematics.

Today, no one would ever think about simulating Maxwell’s equations for the full chip or even the circuit graph level. Rather, the argument for correctly building the chip is an inductive one of the following form:

- (1) Components are exhaustively verified and assumed to be correct at higher levels of abstraction.
- (2) Separation is imposed by design (separation) and validated through global design tools.
- (3) Interconnect is the only device which can connect components through design.
- (4) The collection of components and interconnect is validated at a higher level.
- (5) This process is recursively performed at higher levels of abstraction.

Today, there is no clear idea of abstraction levels in cyber-physical simulations nor are there techniques for divide-and-conquer to enable scaling. Two opportunities which offer hope are:

- (1) Functional Abstract Interfaces: Building abstract interfaces between the major functional aspects of the AV stack (sensing, perception, location, etc.) would seem to offer some ability to do divide-and-conquer. This is not generally carried out today, and the approach of “end-to-end” AI stacks philosophically runs against these sorts of decompositions.
- (2) Execution Semantics: As Figure 1 shows, the cyber-physical paradigm consists of four layers of functionality. The inner core, layer 4, is of course the world of physics which has all the nice execution properties. Layer 3 consists of the traditional actuation and edge sensing functionality which maintains nice physical properties. As we go to layer 2, there is a combination of software and AI which operate in the digital world. Finally, the outer design for the experiment V&V layer has the unique challenge of testing a system with fundamentally physical properties but doing so through a layer dominated by digital properties. Research on performing intelligent test generation which permeates through layer 2 could help break the logjam of current issues.

5. Conclusions

Autonomy sits at the collision of several large communities which previously operated separately. One such community comes from the world of safety critical mechanical systems, another from traditional information technology, and most recently, the new community around artificial intelligence. With autonomy, these three massive systems must interact with each other in an intimate fashion. However, even communication across these vast fields is difficult, and this is especially the case of implicit assumptions within each field.

In this perspective article, the authors have made an attempt to make explicit many of the important underlying assumptions in each field, current gaps in V&V for an autonomy product, and some insight, inspired by deep knowledge from semiconductor V&V, on

promising research directions. In addition, the authors support this work based on open-source PolyVerif environment as a quick testbench for accelerating research productivity.

Funding: This research is partly supported by Horizon Europe project No. 101192749 “Extended zero-trust and intelligent security for resilient and quantum-safe 6G networks and services” and Erasmus+ project No. 2024-1-EE01-KA220-HED-000245441 “Harmonizations of Autonomous Vehicle Safety Validation and Verification for Higher Education”.

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: Author Mahesh Menase was employed by the company Acclivis Technologies Pvt Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Governance and Verification & Validation

In modern societies, products function within a structured legal governance system designed to balance consumer benefit with potential harm, giving rise to the principle of product liability. Although the specifics of liability law vary across jurisdictions, the underlying tenets rest on the dual concepts of expectation and harm. Expectation, typically evaluated through the lens of “reasonable behavior given the totality of the facts,” determines whether a product’s behavior aligns with what an ordinary person would foresee. For instance, it is reasonably expected that a train cannot stop instantaneously if someone stands in its path, whereas expectations regarding autonomous vehicles remain less clearly defined due to evolving technological capabilities. The notion of harm, conversely, establishes the severity and consequence of a product’s failure, distinguishing between domains of differing societal risk; recommendation algorithms for entertainment media are not subjected to the same scrutiny as safety-critical autonomous systems. The overarching governance framework for liability is constructed through legislative action and regulatory development, and its interpretation is refined through judicial proceedings in which courts adjudicate specific factual contexts. To ensure stability and predictability, legal precedents emerging from appellate decisions serve as interpretive anchors that guide future applications of liability principles across cases and technologies [2,25,41].

From a product development standpoint, the convergence of laws, regulations, and legal precedents establishes the overarching governance framework [25] within which system specifications must be defined and implemented [42]. This framework not only constrains design choices but also shapes the assurance processes required to demonstrate compliance and societal trustworthiness. Within this context, validation refers to the process of confirming that the product design fulfills the intended user needs and operational requirements, while verification ensures that the system has been constructed correctly in accordance with its design specifications. Together, these complementary activities form the technical foundation of product assurance, linking engineering rigor with the broader legal and regulatory expectations governing product safety and performance.

The Master Verification and Validation (MaVV) process serves as the overarching mechanism through which a product must demonstrate that it has been reasonably tested in light of its potential to cause harm. In essence, MaVV operationalizes the principle of **reasonable due diligence** by linking engineering assurance activities to the legal expectation of safety and accountability. It does so through three foundational concepts [43]:

1. **Operational Design Domain (ODD):** Defines the environmental conditions, operational scenarios, and system boundaries within which the product is intended to function safely and effectively.
2. **Coverage:** Represents the degree to which the product has been validated across the entirety of its ODD, thereby quantifying the completeness of testing.

3. **Field Response:** Specifies the mechanisms and procedures through which design shortcomings are identified, analyzed, and corrected following field incidents to prevent recurrence and mitigate future harm.

As illustrated in Figure A1, the Verification and Validation (V&V) process provides a primary input into the broader governance structure that determines liability. Within this structure, each component of the MaVV framework must demonstrate “reasonable due diligence.” For instance, it would be unreasonable for an autonomous vehicle to disengage control mere milliseconds before an impending collision, as such an operational boundary would violate the expectation of responsible system design.

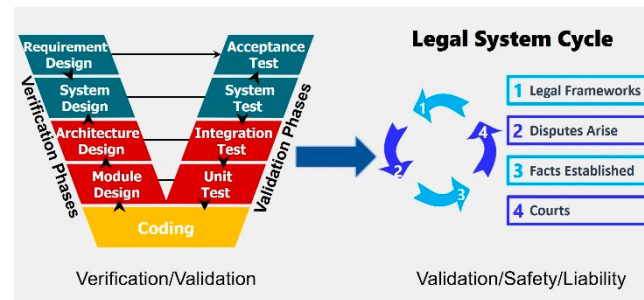


Figure A1. V&V and Governance Framework.

In practical engineering terms, MaVV is implemented through a set of *Minor V&V* (MiVV) processes that collectively constitute the operational workflow:

1. **Test Generation:** From the specified ODD, relevant test scenarios are generated to exercise system functionality across the intended operational envelope.
2. **Execution:** Each test is executed on the product under development, representing a functional transformation that produces measurable outputs.
3. **Criteria for Correctness:** The resulting outputs are evaluated against explicit criteria that define success or failure, ensuring objective assessment of conformance.

Each of these steps carries substantial complexity and cost. Because the ODD often encompasses a vast state space, intelligent and efficient test generation is critical. Early phases of testing are typically performed manually, but such methods rapidly become inefficient as system complexity scales. In virtual execution environments, pseudo-random directed methods are often employed to improve coverage efficiency, while symbolic or formal methods may be used in constrained situations to mathematically propagate large state spaces through design analysis. Although symbolic approaches offer theoretical completeness, they are often limited by computational intractability, as many underlying algorithms are NP-Complete.

Test execution can be conducted either physically or virtually. Physical testing, while accurate by definition, tends to be slow, costly, and often limited in controllability and observability—posing additional risks for safety-critical systems. Conversely, virtual testing provides significant advantages in speed, cost, controllability, and safety, and enables verification and validation well before physical prototypes exist. This early-stage testing capability is represented in the classic V-model (Figure 1), which captures the iterative relationship between design and test phases. However, because virtual methods rely on abstractions of reality, they inherently introduce modeling inaccuracies. To balance fidelity and efficiency, hybrid approaches such as Software-in-the-Loop (SiL) and Hardware-in-the-Loop (HiL) are often employed, combining the scalability of simulation with the realism of physical components.

The determination of correctness relies on comparing observed outcomes against either a golden model or an anti-model. A golden model typically serves as an independently

verified reference—often virtual—whose results can be directly compared with those of the product under test. Such approaches are common in computer architecture domains (e.g., ARM, RISC-V) where validated behavioral models serve as benchmarks. Nonetheless, mismatches in abstraction levels between the golden model and the actual product must be carefully managed. In contrast, the anti-model approach defines error states that the product must never enter; thus, correctness is established by demonstrating that the system operates entirely outside of these forbidden states. In the context of autonomous vehicles, for example, error states could include collisions, traffic-law violations, or breaches of safety constraints.

The Master Verification and Validation (MaVV) framework ultimately seeks to establish a defensible argument that the product has been reasonably and comprehensively tested over its defined Operational Design Domain (ODD). To this end, MaVV involves the systematic construction of a database capturing the results of exploratory analyses, simulations, and test executions across the ODD state space. This accumulated evidence forms the empirical basis for assessing coverage and constructing an argument for completeness. Such arguments are inherently probabilistic in nature, acknowledging that exhaustive testing of all possible operational states is infeasible. Instead, the goal is to demonstrate—with quantifiable confidence—that the product has been validated over a representative and sufficiently broad subset of operational conditions to meet the expectation of reasonable safety.

Once the product enters operational deployment, field performance provides an additional feedback channel. Each observed failure or field return is subjected to diagnostic analysis to identify causal factors and determine why the original verification process did not detect the issue. These post-deployment analyses serve a critical role in strengthening the MaVV framework by driving continuous improvement—updating the test methodology, expanding scenario coverage, and refining the underlying models to prevent recurrence of similar failures in future iterations. Figure A2 outlines this broader flow.

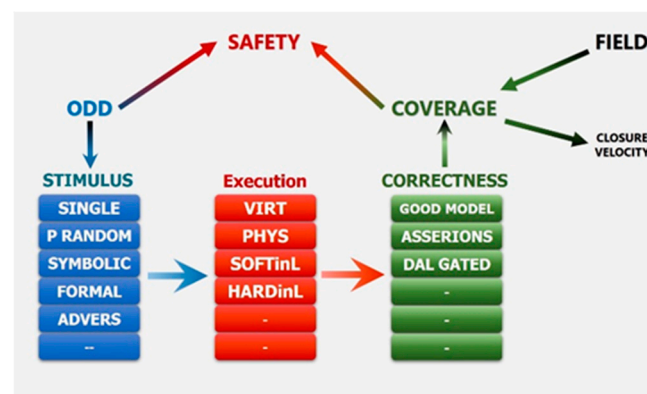


Figure A2. MaVV and MiVV V&V Process.

Appendix B. Example of Traditional ISO 26262

As an example, the flow for validating the braking system in an automobile through ISO 26262 would have the following steps:

- (1) Define Safety Goals and Requirements (Concept Phase): Hazard Analysis and Risk Assessment (HARA): Identify potential hazards related to the braking system (e.g., failure to stop the vehicle, uncommanded braking). Assess risk levels using parameters like severity, exposure, and controllability. Define Automotive Safety Integrity Levels (ASIL) for each hazard (ranging from ASIL A to ASIL D, where D is the most

- stringent). Define safety goals to mitigate hazards (e.g., ensure sufficient braking under all conditions).
- (2) **Develop Functional Safety Concept:** Translate safety goals into high-level safety requirements for the braking system. Ensure redundancy, diagnostics, and fail-safe mechanisms are incorporated (e.g., dual-circuit braking or electronic monitoring).
 - (3) **System Design and Technical Safety Concept:** Break down functional safety requirements into technical requirements, design the braking system with safety mechanisms like Hardware (e.g., sensors, actuators). Software (e.g., anti-lock braking algorithms). Implement failure detection and mitigation strategies (e.g., failover to mechanical braking if electronic control fails).
 - (4) **Hardware and Software Development:** Hardware Safety Analysis (HSA): Validate that components meet safety standards (e.g., reliable braking sensors). Software Development and Validation: Use ISO 26262-compliant processes for coding, verification, and validation. Test braking algorithms under various conditions.
 - (5) **Integration and Testing:** Perform verification of individual components and subsystems to ensure they meet technical safety requirements. Conduct integration testing of the complete braking system, focusing on: Functional tests (e.g., stopping distance), Safety tests (e.g., behavior under fault conditions), and Stress and environmental tests (e.g., heat, vibration).
 - (6) **Validation (Vehicle Level):** Validate the braking system against safety goals defined in the concept phase. Perform real-world driving scenarios, edge cases, and fault injection tests to confirm safe operation. Verify compliance with ASIL-specific requirements.
 - (7) **Production, Operation, and Maintenance:** Ensure production aligns with validated designs, implement operational safety measures (e.g., periodic diagnostics, maintenance), monitor and address safety issues during the product's lifecycle (e.g., software updates).
 - (8) **Confirmation and Audit:** Use independent confirmation measures (e.g., safety audits, assessment reviews) to ensure the braking system complies with ISO 26262.

Appendix C. Electronics Megatrends and Progression of Computing

The evolution of electronics and computing technologies fundamentally transformed industrial and organizational processes across the twentieth century. The advent of centralized computing systems, pioneered by firms such as IBM and Digital Equipment Corporation (DEC), revolutionized the management of information and operations within large enterprises. These systems enabled unprecedented levels of productivity and process automation, particularly in domains such as finance, human resources, and administrative management, where digital record-keeping and computational analysis supplanted labor-intensive paper-based workflows. The resulting efficiency gains not only redefined internal business operations but also laid the technological foundation for subsequent waves of distributed and networked computing architectures.

The next transformative wave in economy-shaping technologies emerged with the proliferation of edge computing devices—such as personal computers, smartphones, and tablets (illustrated in red in Figure A3). These platforms decentralized computational power, bringing intelligence and connectivity directly to end users. Leveraging this new technological layer, companies such as Apple, Amazon, Facebook, and Google catalyzed a paradigm shift in global productivity, particularly in advertising, distribution, and customer engagement. For the first time, businesses could reach individual consumers instantaneously and at global scale, effectively erasing traditional geographic and logistical barriers.

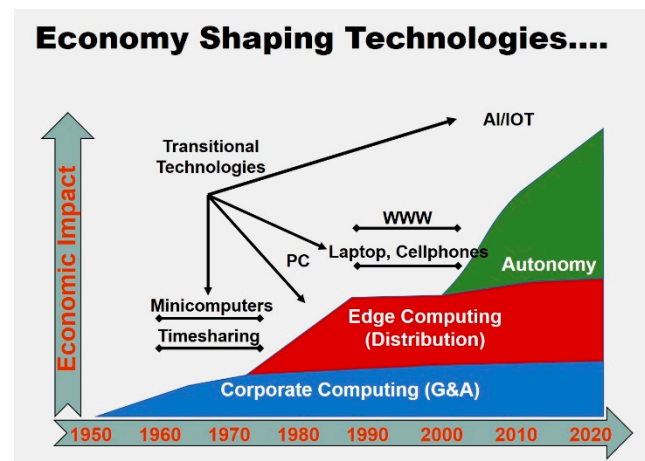


Figure A3. Electronics Megatrends.

This edge-driven digital transformation disrupted multiple sectors of the economy, including education (through online learning), retail (via e-commerce), entertainment (through streaming media), commercial real estate (via virtualization of workplaces), and healthcare (through telemedicine and remote monitoring). As this wave matures, a new frontier is emerging—characterized by the dynamic integration of electronic intelligence with physical assets, enabling the rise of cyber-physical systems and, ultimately, autonomous operation across industrial and societal domains.

As illustrated in Figure A4, the evolution of electronic systems reflects a progressive abstraction of functional construction. The initial stage centered on hardware-based or pseudo-hardware implementations, utilizing fixed physical designs or reconfigurable logic such as Field-Programmable Gate Arrays (FPGAs) and microcoded control. The subsequent stage introduced the concept of processor architectures, which enabled system functionality to be expressed through software rather than hardwired circuitry. In this paradigm, system behavior was encoded as human-authored design artifacts written in standardized programming languages such as C or Python.

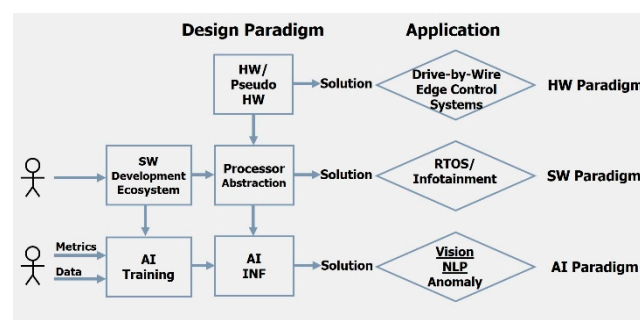


Figure A4. Progression of System Specification (HW, SW, AI).

The emergence of the processor abstraction marked a revolutionary shift in design flexibility: it allowed system functionality to evolve without requiring modifications to the underlying physical assets. However, this flexibility came with a significant demand for human labor, as large teams of programmers were needed to develop and maintain increasingly complex software systems. In the contemporary era, the advent of Artificial Intelligence (AI) represents the next major inflection point in this progression. AI enables the construction of functional software not purely through manual programming but through the integration of models, data, and performance metrics, allowing systems to

learn and adapt their behavior through statistical inference and optimization rather than explicit human coding.

References

- Othman, K. Exploring the implications of autonomous vehicles: A comprehensive review. *Innov. Infrastruct. Solut.* **2022**, *7*, 165–196. [CrossRef]
- Razdan, R. *Unsettled Technology Areas in Autonomous Vehicle Test and Validation* EPR2019001; SAE International: Warrendale, PA, USA, 2019.
- Razdan, R. *Product Assurance in the Age of Artificial Intelligence*; SAE International Research Report EPR2025011; SAE International: Warrendale, PA, USA, 2025. [CrossRef]
- ISO 26262; Road Vehicles Functional Safety. International Organization for Standardization: Geneva, Switzerland, 2018. Available online: <https://www.iso.org/publication/PUB200262.html> (accessed on 1 November 2025).
- Moore, G.E. Cramming More Components onto Integrated Circuits. *Electronics* **1965**, *38*, 114–117. [CrossRef]
- Lavagno, L.; Markov, I.L.; Martin, G.; Scheffer, L.K. (Eds.) *Electronic Design Automation for Integrated Circuits Handbook*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2022; Volume 2, ISBN 978-1-0323-3998-6.
- Wang, L.-T.; Chang, Y.-W.; Cheng, K.-T. (Eds.) *Electronic Design Automation: Synthesis, Verification, and Test*; Morgan Kaufmann/Elsevier: Amsterdam, The Netherlands, 2009; ISBN 978-0-12-374364-0.
- Jansen, D. (Ed.) *The Electronic Design Automation Handbook*; Springer: Berlin/Heidelberg, Germany, 2010; ISBN 978-1-4419-5369-8.
- AS9100; Quality Systems—Aerospace—Model for Quality Assurance in Design, Development, Production, Installation and Servicing. SAE International: Warrendale, PA, USA, 1999. Available online: <https://www.sae.org/standards/content/as9100/> (accessed on 1 November 2025).
- TÜV SÜD. IECEx International Certification Scheme—Testing & Inspection. Available online: <https://www.tuvsud.com/en-us/services/product-certification/iecex-international-certification-scheme> (accessed on 1 November 2025).
- Dassault Aviation. Dassault Aviation, a Major Player to Aeronautics. 2025. Available online: <https://www.dassault-aviation.com/en/> (accessed on 1 November 2025).
- Siemens. Siemens Digital Industries Software. Simcenter Tire Software. Available online: <https://plm.sw.siemens.com/en-US/simcenter/mechanical-simulation/tire-simulation> (accessed on 19 March 2025).
- Price, D. Pentium FDIV flaw—Lessons learned. *IEEE Micro* **1995**, *15*, 86–88. [CrossRef]
- CMMI Product Team. *CMMI for Development, Version 1.3*; Technical Report CMU/SEI-2010-TR-033; Carnegie Mellon University, Software Engineering Institute’s Digital Library: Pittsburgh, PA, USA, 2010. [CrossRef]
- Hellebrand, S.; Rajski, J.; Tarnick, S.; Courtois, B.; Figueras, J. *Random Testing of Digital Circuits: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 1993. [CrossRef]
- ISO/PAS 21448:2019; Road Vehicles—Safety of the Intended Functionality. International Organization for Standardization: Geneva, Switzerland, 2019.
- DO-178C; Software Considerations in Airborne Systems and Equipment Certification. RTCA Incorporated: Washington, DC, USA, 2011. Available online: <https://www.rtca.org/training/do-178c-training/> (accessed on 1 November 2025).
- Varshney, K.R. Engineering Safety in Machine Learning. In Proceedings of the 2016 Information Theory and Applications Workshop (ITA), La Jolla, CA, USA, 31 January–5 February 2016; pp. 1–5. [CrossRef]
- Schwalbe, G.; Schels, M. A Survey on Methods for the Safety Assurance of Machine Learning Based Systems. In *Computer Safety, Reliability, and Security (SAFECOMP 2020)*; LNCS 12234; Springer: Berlin/Heidelberg, Germany, 2020; pp. 263–275. [CrossRef]
- Batarseh, F.A.; Freeman, L.; Huang, C.H. A survey on artificial intelligence assurance. *J. Big Data* **2021**, *8*, 60. [CrossRef]
- ISO/PAS 8800:2024; Road Vehicles—Safety and Artificial Intelligence. International Organization for Standardization (ISO): Geneva, Switzerland, 2024. Available online: <https://www.iso.org/standard/83303.html> (accessed on 1 November 2025).
- Renz, J.; Oehm, L.; Klusch, M. Towards Robust Training Datasets for Machine Learning with Ontologies: A Case Study for Emergency Road Vehicle Detection. *arXiv* **2024**, arXiv:2406.15268. [CrossRef]
- Kouvaros, P.; Leofante, F.; Edwards, B.; Chung, C.; Margineantu, D.; Lomuscio, A. Verification of semantic key point detection for aircraft pose estimation. In Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, Rhodes, Greece, 2–8 September 2023. [CrossRef]
- IEEE Std 2846-2022; IEEE Standard for Assumptions in Safety-Related Models for Automated Driving Systems. IEEE: Piscataway, NJ, USA, 2022.
- National Institute of Standards and Technology (NIST). AI Risk Management Framework (AI RMF 1.0). U.S. Department of Commerce, 2023. Available online: <https://www.nist.gov/itl/ai-risk-management-framework> (accessed on 1 November 2025).

26. Koopman, P.; Widen, W.H. A Reasonable Driver Standard for Automated Vehicle Safety. Legal Studies Research Paper No. 4475181; University of Miami School of Law. Available online: https://www.researchgate.net/publication/373919605_A_Reasonable_Driver_Standard_for_Automated_Vehicle_Safety (accessed on 8 October 2025).
27. Insurance Institute for Highway Safety. First Partial Driving Automation Safeguard Ratings Show Industry Has Work to Do. March 2024. Available online: <https://www.iihs.org/news/detail/first-partial-driving-automation-safeguard-ratings-show-industry-has-work-to-do> (accessed on 1 November 2025).
28. Insurance Institute for Highway Safety. IIHS Creates Safeguard Ratings for Partial Automation. IIHS-HLDI Crash Testing and Highway Safety. 2022. Available online: <https://www.iihs.org/news/detail/iihs-creates-safeguard-ratings-for-partial-automation> (accessed on 1 November 2025).
29. National Highway Traffic Safety Administration. Final Rule: Automatic Emergency Braking Systems for Light Vehicles. April 2024. Available online: https://www.nhtsa.gov/sites/nhtsa.gov/files/2024-04/final-rule-automatic-emergency-braking-systems-light-vehicles_web-version.pdf (accessed on 1 November 2025).
30. Uvarov, T.; Tripathi, B.; Fainstain, E.; Tesla Inc. Data Pipeline and Deep Learning System for Autonomous Driving. US11215999B2, 20 June 2018. Available online: <https://patents.google.com/patent/US11215999B2/en> (accessed on 1 November 2025).
31. Tesla. *Tesla Dojo Supercomputer: Training AI for Autonomous Vehicles*; Tesla: Austin, TX, USA, 2021.
32. Winner, H.; Lemmer, K.; Form, T.; Mazzega, J. *PEGASUS—First Steps for the Safe Introduction of Automated Driving*; Springer Nature: Berlin/Heidelberg, Germany, 2019; pp. 185–195. [CrossRef]
33. The Global Initiative for Certifiable AV Safety. Available online: <https://safetypool.ai> (accessed on 1 November 2025).
34. ASAM, e.V. ASAM OpenSCENARIO® DSL. 2024. Available online: <https://www.asam.net/standards/detail/openscenario-dsl/> (accessed on 1 November 2025).
35. *UL 4600*; Standard for Safety for the Evaluation of Autonomous Products. (3rd ed.). UL Standards & Engagement: Evanston, IL, USA, 2023.
36. Diaz, M.; Woon, M. *DRSPI—A Framework for Preserving Automated Vehicle Safety Claims by Unknown Unknowns Recognition and Dynamic Runtime Safety Performance Indicator Improvement*; SAE Technical Papers on CD-ROM/SAE Technical Paper Series; SAE International: Warrendale, PA, USA, 2022. [CrossRef]
37. Goldberg, D. What every computer scientist should know about floating-point arithmetic. *ACM Comput. Surv.* **1991**, *23*, 5–48. [CrossRef]
38. Nagel, L.W.; Pederson, D.O. *SPICE (Simulation Program with Integrated Circuit Emphasis)*; Technical Report No. UCB/ERL M382; EECS Department, University of California: Berkeley, CA, USA, 1973.
39. University of California, Berkeley. *SPICE3F5: Simulation Program with Integrated Circuit Emphasis*; Electronics Research Laboratory, University of California: Berkeley, CA, USA, 1993.
40. Shannon, C.E. A symbolic analysis of relay and switching circuits. *Trans. Am. Inst. Electr. Eng.* **1938**, *57*, 713–723. [CrossRef]
41. Razdan, R. *Unsettled Topics Concerning Automated Driving Systems and the Transportation Ecosystem* EPR2019005; SAE International: Warrendale, PA, USA, 2019.
42. Ross, K. *Product Liability Law and Its Effect on Product Safety*; In Compliance Magazine: Littleton, MA, USA, 2023. Available online: <https://incompliancemag.com/product-liability-law-and-its-effect-on-product-safety/> (accessed on 1 November 2025).
43. Forsberg, K.; Mooz, H. The relationship of system engineering to the project cycle. In Proceedings of the First Annual Symposium of National Council on System Engineering, Chattanooga, TN, USA, 20–23 October 1991; Volume 1, pp. 57–65. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.