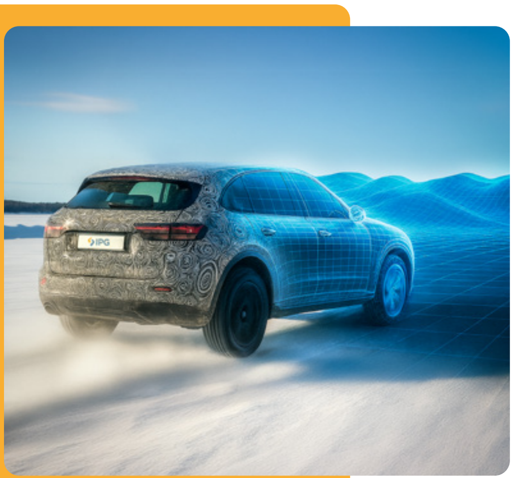
Solutions for Virtual Vehicle Development

The Evolution of Testing in Automotive Development – From Validation Bottlenecks to Strategic Enablers

Table of Contents



The Evolution of Testing in Automotive Development – From Validation Bottlenecks to Strategic Enablers

Author:
Dr. Alexander Ahlert, IPG Automotive GmbH

1. Executive Summary

The automotive industry is undergoing a fundamental transformation. Vehicles are becoming increasingly software-defined, electrified, and connected, bringing unprecedented complexity to development processes. In this landscape, the traditional view of testing as a late-stage quality gatekeeper is no longer sufficient.

This white paper explores the evolution of testing strategies in automotive development, from physical, sequential methods to integrated, model-based and increasingly virtual approaches. Modern testing is not just about catching defects. It is about enabling faster innovation, ensuring compliance, and supporting safe as well as continuous delivery of functionalities, while saving time and costs.

The white paper outlines how industry leaders are rethinking their validation strategies to address growing E/E and software complexity, accelerate development cycles, and build resilience in increasingly agile environments. It explores the transformation toward validation "shifted left" (e.g. MIL, SIL and Rapid HIL) and the strategic importance of test automation and data management.

Finally, the organizational implications of this transformation are highlighted: new skills, new roles, and a cultural shift in how testing is perceived and performed.

This white paper addresses decision-makers, engineering leaders or systems engineers and aims to inspire a strategic perspective on testing, not just as a cost factor or hurdle, but as a competitive advantage in the future of mobility.

2. Introduction: From Quality Gatekeeper to Innovation Accelerator

Testing has long been a cornerstone of automotive development and a necessary gatekeeper to ensure vehicle quality, safety, and compliance. Traditionally, this meant physical testing on test benches, proving grounds, and on public roads, often conducted late in the development cycle after the integration of all relevant components and subsystems in a full vehicle.

While effective in their time (hardware-driven), these methods are increasingly misaligned with the demands of today's vehicles. Software-defined architectures, electrification, automated driving systems, and over-the-air (OTA) updates have radically altered the product lifecycle. Testing must now keep pace with software development speed, cross-domain dependencies, and compressed time-to-market expectations.

The result is a paradigm shift: Testing is no longer a final phase, but an ongoing, integrated process that spans the entire development process and even product lifecycle after SOP. It starts earlier, happens more often and is increasingly virtual. Validation and holistic test strategies are evolving to detect issues sooner, reduce reliance on physical prototypes and hardware, and enable collaborative, digitalized development.

This white paper traces the evolution of testing strategies in the automotive sector. It examines the limitations of legacy practices, the drivers of change, and the emerging methods that are reshaping how validation is planned, executed and leveraged, both in terms of technology and organization.

The aim is to present a new perspective on testing: not as a bottleneck, but as a strategic enabler of innovation, quality, and speed in a fast-moving industry.

3. The Legacy Approach: Physical, Hardware-Centric Testing

For decades, the automotive industry relied on physical testing as the backbone of its validation strategy. Component-level tests, system integration test benches and vehicle prototypes formed a robust but resource-heavy infrastructure to ensure product reliability, safety, and compliance.

This traditional approach was deeply rooted in the sequential V-model of mechatronic system development: Design and implementation occurred first, followed by integration and finally testing, which often took place late in the process and with many full vehicle prototypes. At each stage, feedback loops were long and the detection of issues late in the cycle could lead to costly delays or restricting compromises, see also Fig. 01.

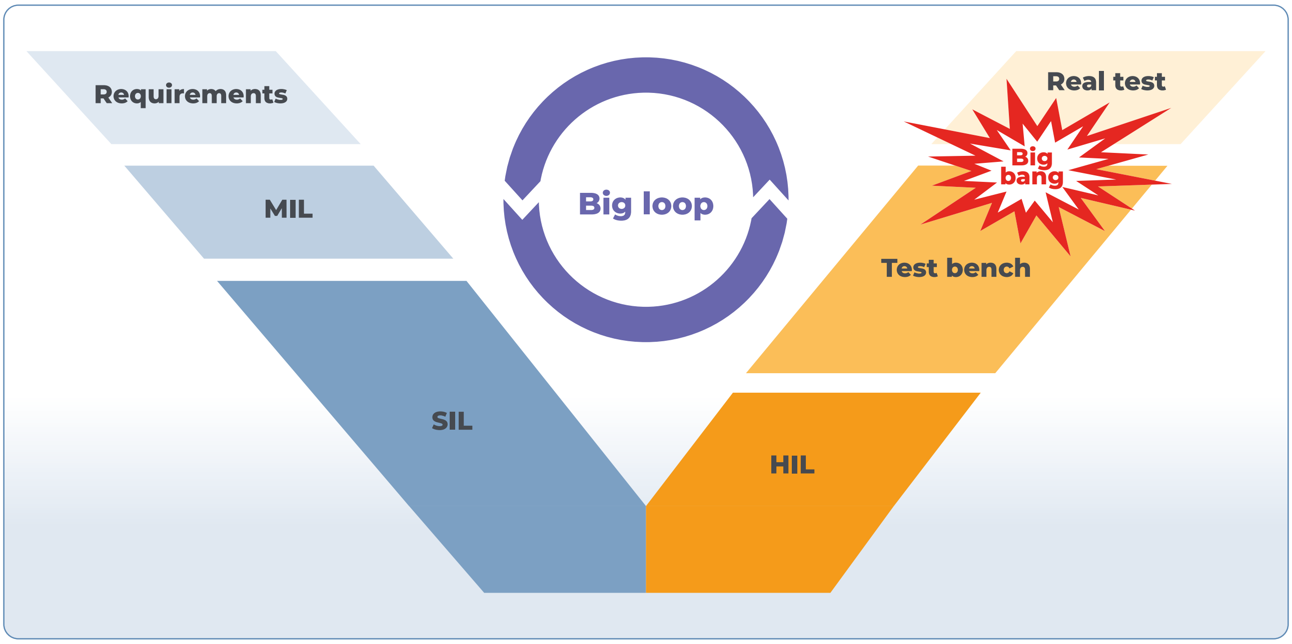


Fig. 01: Legacy integration and testing approach in the V-model

Physical test benches and vehicle prototypes offered realism and subjective evaluation capabilities for experienced engineers, but lacked flexibility and scalability. They were often constrained by availability, hardware lead times, limited reusability, and logistical complexity. Additionally, debugging software issues discovered during late-stage integration was time-consuming, cumbersome, and frequently required coordination across several engineering teams.

Key characteristics of this legacy testing approach included:

- Late-stage validation: Most tests occurred after software and hardware integration, often when design flexibility was already low
- Limited scalability: Physical test assets were expensive, lacked reproducibility, and could not easily accommodate large numbers of test cases or variants
- Siloed execution: Testing was often isolated by domain (e.g. powertrain, vehicle dynamics, ADAS, HMI, etc.), limiting cross-functional insights
- Manual processes: Setup, execution and reporting relied heavily on manual workflows, resulting in low efficiency and risk of human error

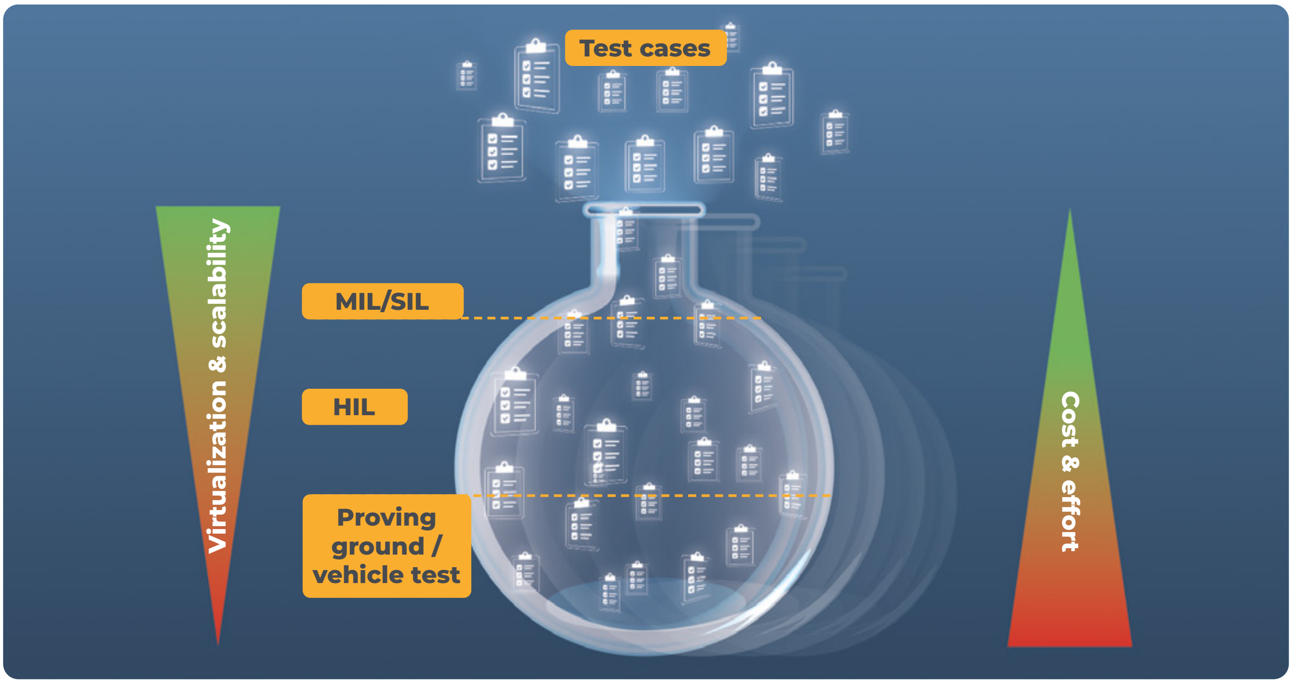


Fig. 02: Classic test distribution in hardware-centric validation

In an era when product development was largely mechanical and software feature sets were relatively isolated and static, these limitations were manageable. But as vehicle complexity increased exponentially (see Fig. 03.), especially with the rise of software, new E/E architectures and connectivity, fundamental problems in this approach began to arise.

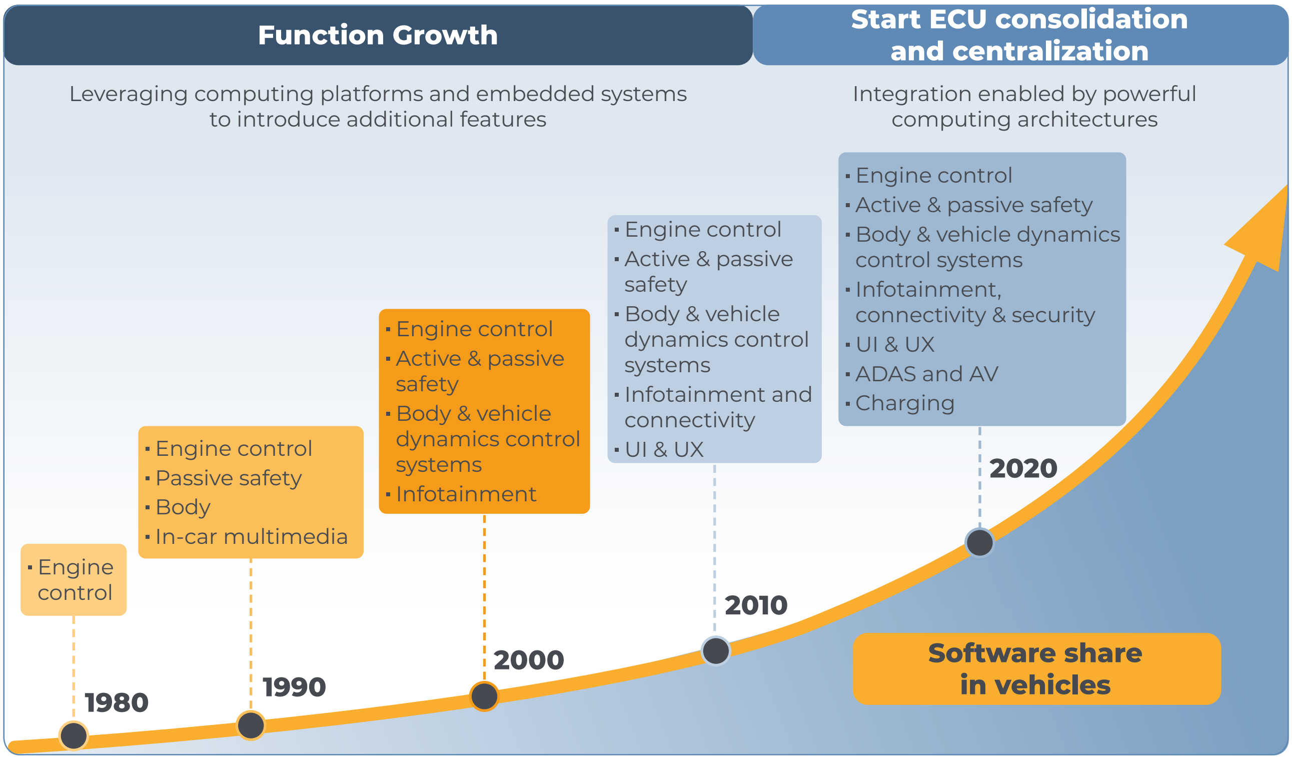


Fig. 03: Exponential rise of software share and complexity in vehicle

Today, these legacy practices are insufficient to keep pace with the speed, scale, and interconnectivity demanded by modern vehicles. The shift toward a more agile, model-driven and software-centric world requires a fundamentally different testing paradigm, one that begins earlier, scales faster, and is deeply integrated across the development lifecycle.

4. Why Change Was Needed: Pressure Points and Industry Drivers

The evolution of testing strategies in automotive development was not optional; it was driven by necessity. With the mentioned modern vehicles, legacy testing approaches struggled to keep up with growing complexity, cross-domain applications, new regulatory demands, and faster development cycles.

Several key pressure points pushed the industry to rethink how testing is performed:

1) Software complexity and functional integration

Modern vehicles can contain over 100 million lines of code, distributed across dozens of interconnected control units in different E/E architectures (central, domain, zonal, etc.). Features are no longer isolated, they are interdependent across domains: powertrain, ADAS, vehicle dynamics control, HMI, body, infotainment, and connectivity. Testing isolated systems is no longer sufficient; holistic validation along e.g. effect chains and customer journeys is required.

Moreover, frequent software updates and dynamic feature sets demand validation capabilities that are fast, repeatable, and traceable throughout the lifecycle.

2) Compressed time to market and rapid software validation

Competitive pressure and costumer expectations have dramatically reduced acceptable development times. OEMs now strive to launch new vehicle platforms or major updates in shorter cycles than ever before. Especially in China, vehicle development times are drastically reduced, in some cases up to 18 to 24 months.

This requires scaled testing strategies that shift validation activities to earlier stages in the development process (a key part of shift left) and support faster iterations. Late discovery of errors in physical testing stages is no longer acceptable. In contrast, robust and scalable software solutions, that create competitive advantages and the ability to fix software issues as quickly as possible (including the necessary debugging, validation, and OTA updates), are mandatory to fulfil customer expectations, especially in China and the US.

3) Costs and resource constraints

Physical testing infrastructures are expensive to build, maintain, and scale. They require physical prototypes, sometimes complex test benches with e.g. environmental chambers, and specialized teams, all with high associated costs.

In contrast, virtual testing offers repeatability, scalability, and parallelization with significantly lower marginal cost. Virtual methods enable testing earlier and more often, without the need for physical hardware.

4) Regulatory demands and safety assurance

The increasing complexity of safety-critical systems (e.g. automated driving and fail-operational systems like steer-by-wire) has elevated the need for comprehensive, scenario-based, and standard-compliant testing.

Functional safety (ISO 26262), SOTIF (ISO 21448), cybersecurity (ISO/SAE 21434), and UNECE regulations (e.g. R157) require systematic validation, transparency and traceability, which are difficult to achieve with approaches focused on manual, physical testing.

5) Agile and continuous development concepts

Software-defined and connected vehicles force the automotive industry to shift toward agile, software-centric workflows, continuous integration (CI) and even continuous deployment (CD). This is especially relevant for technology-led and entertainment-driven markets, where OTA updates are becoming the rule.

Such concepts require continuous testing (CT), not just at release milestones, but throughout development. This requires virtualization and simulation environments, automation, and seamless integration into CI/CD pipelines with virtual fleets and sophisticated test management, i.e. a continuous development paradigm with incorporated CI/CT/CD, sometimes called automotive DevOps.

In short, legacy testing and development practices, designed for the embedded mechatronic system development era, can no longer address the speed, scale and complexity of today's automotive development. The shift to more adaptive, virtualized testing and validation approaches is not just a strategic choice, but a technical and economic imperative.

5. The Modern Testing Strategy: Shifting Left, Scaling Up and Bridging Domains

In response to the growing challenges relating to complexity, costs, and development speed, the automotive industry is undergoing a strategic shift in how testing is addressed. This modern approach is characterized by three key strategies:

1. Shifting testing and validation to earlier stages of the development process (shift left)
2. Scaling up validation through virtualization and automation
3. Integrating testing holistically across domains and development stages by using inter-connected, layered testing environments that do not break the toolchain

Thus, virtualization and simulation are key enablers of the modern testing strategy. By simulating mechatronic systems such as vehicles (consisting of sensors, actors, and controls/logics) with virtual drivers in virtualized generic or real-world scenarios, development teams can execute extensive test catalogues long before prototypes exist. However, it is important to emphasize that the simulation and testing toolchains must be designed in such a way that these models can be reused and combined with real physical components as soon as they are available. These toolchains must be free of breaks to ensure traceability, comparability, and determinism thus enabling simplified certification and homologation later on.

In the modern approach, testing and validation begin right at the start, at the concept and architecture level: as soon as first requirements are written (requirements validation) and initial control models or drive system topologies are developed (model-in-the-loop), for example. This shift enables simulation- and fact-based decision-making, earlier detection of design errors and platform validation, faster iteration loops as well as more robust concepts, see Fig. 04.

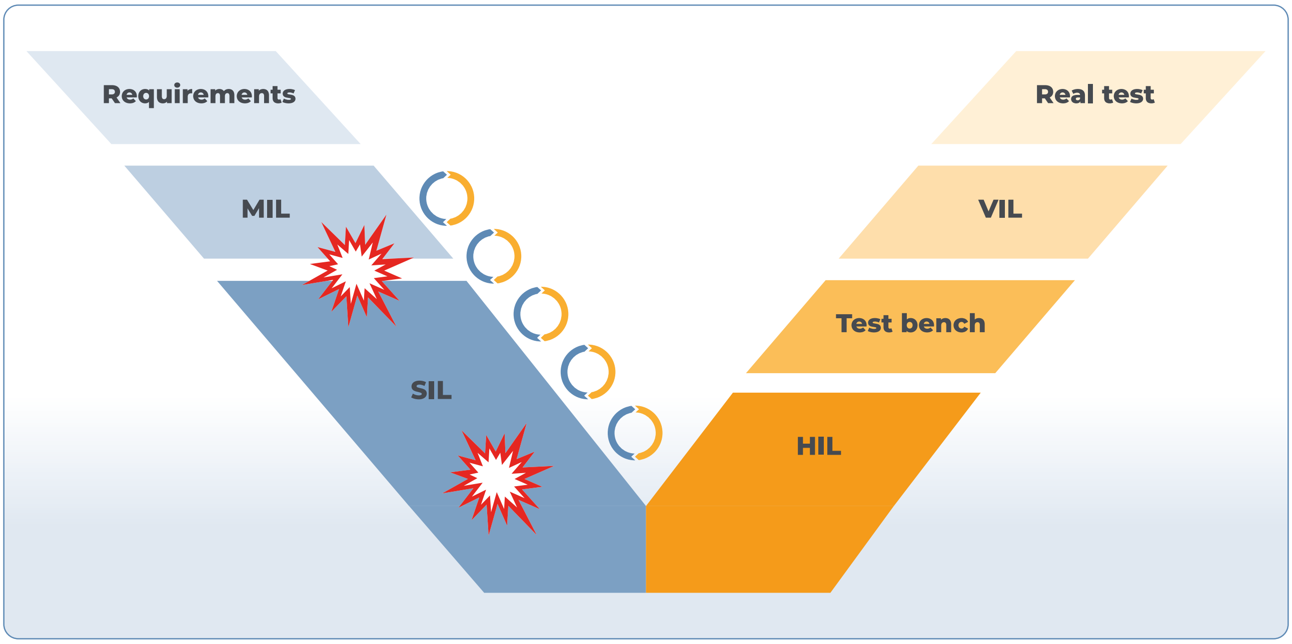


Fig. 04: Improved integration and testing approach in the V-model

The earlier defects are identified, the lower the costs and effort required to resolve them: a well-known principle in classical software development that is now being enabled by powerful simulation tools and automation frameworks for software-in-the-loop (SIL) solutions in the automotive industry. This approach plays a key role in solving the validation challenges of software-defined vehicles (SDVs) and autonomous driving systems. More insights into SIL are given in chapter 8.

However, a virtualized, holistic testing and development strategy involves more than simply shifting activities to earlier stages. It also plans for later-stage activities and test benches such as scaling with MIL and SIL, focusing on communication and real-time dependencies in HIL, including real physics and system dynamics on e.g. component and system test benches (e.g. powertrain- or steering-in-the-loop) up to vehicle-in-the-loop (VIL).

A modern testing approach is best visualized as a layered testing pyramid (Fig. 05), where each layer adds realism, integration depth and validation fidelity:

- Model-in-the-loop (MIL): Testing e.g. control algorithms and system behavior at the model level before any code is generated or hardware is produced. This allows testing the control logic and system architecture from day one.
- Software-in-the-loop (SIL): Verifying and validating production-ready code in a virtual environment in the form of virtual ECUs, independent of target hardware.
- Hardware-in-the-loop (HIL): Integrating real ECUs and hardware with simulation to validate embedded software and signal interfaces including bus protocols under real-time conditions.
- Component and system test benches: Focusing on physical components and subsystems to validate and parametrize simulation models, characterize real system performance, and perform early integration as well as acceptance tests before a full vehicle prototype is available. Additionally, the benefits of laboratory conditions must be highlighted.
- Vehicle-in-the-loop (VIL) and road-to-rig: Connecting full vehicle test benches or test vehicles to digital environments, supporting the end-to-end integration of virtual test scenarios with highest credibility.
- Road and proving ground testing for final validation under real-world conditions with full subjective evaluation capabilities.

These layers provide a scalable testing pyramid, starting with fast, early-stage simulations and progressing to increasingly realistic, albeit more resource-intensive and complex integration stages. By executing thousands of tests across multiple abstraction levels, teams can identify issues earlier, reduce their reliance on physical prototypes, and validate more variants and edge cases. Furthermore, this layered approach allows for strategic use of costly resources such as HIL, test benches, and real vehicles.

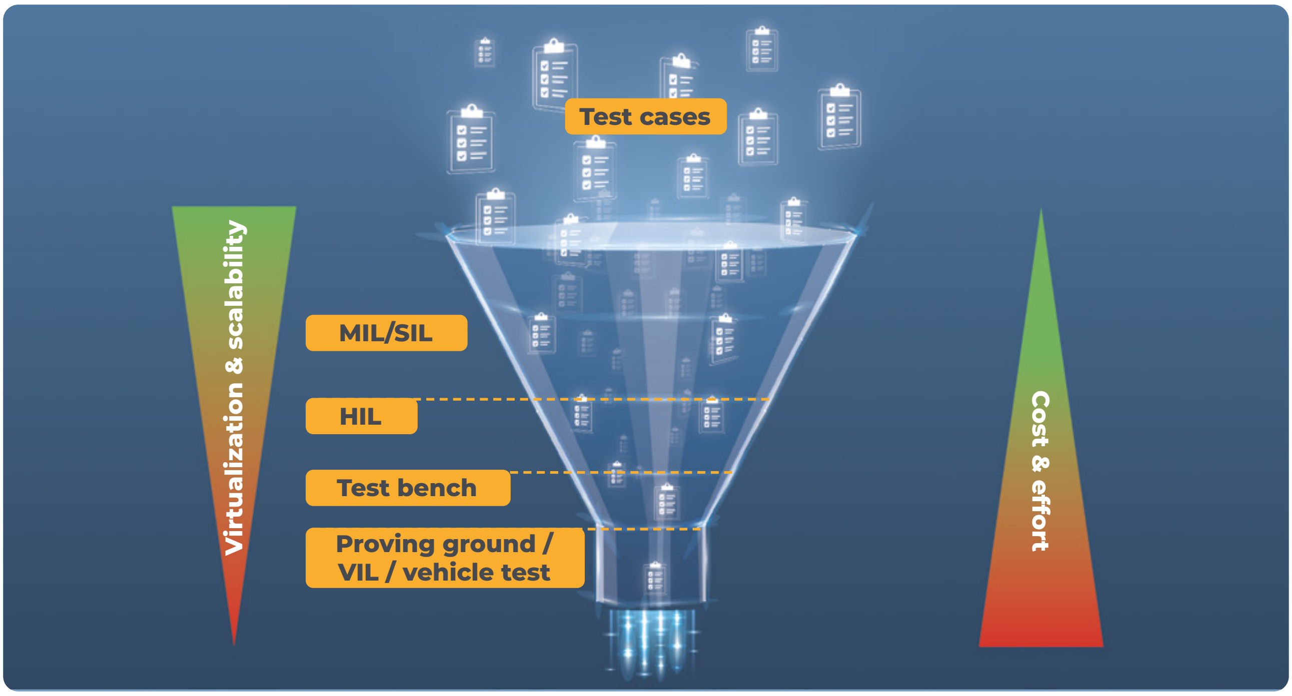


Fig. 05: Modern Vehicle Testing Pyramid

Scalability, reusability, and automation across the testing pyramid are key. Modern testing and validation strategies should aim to scale horizontally by running more tests and incorporating more variants and conditions. They should also integrate vertically by combining virtualization with existing hardware across development stages. This is enabled by:

- Reusable test scenarios and virtual vehicle fleets with traceable parameter sets as well as (sub-)system and component models
- Test orchestration tools that automate execution across platforms (MIL to VIL)
- Unified data management systems and toolchain interoperability

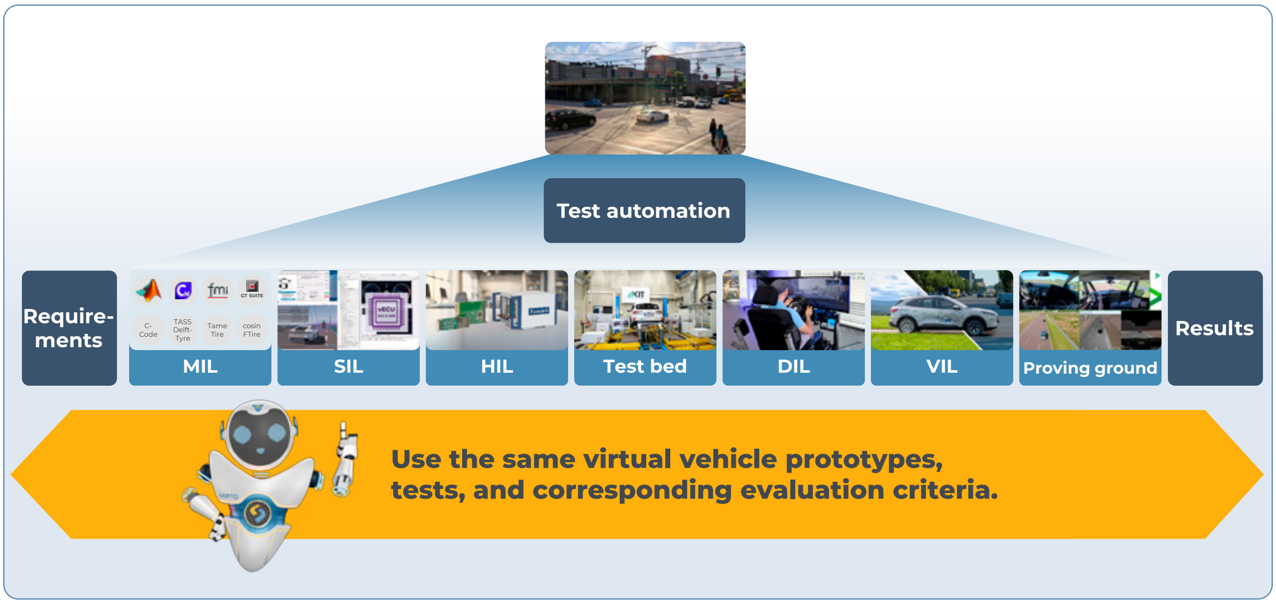


Fig. 06: Continuous Virtual Development Pipeline

However, testing maturity is not just about having the right tools and test environments or simply doing more tests. It is also about orchestrating them intelligently, assigning tests purposefully, and connecting insights across the development process. Therefore, a holistic testing strategy is at the core of the verification and validation process until release and homologation. Selecting the appropriate test level and allocation that is application-specific and purpose-driven is crucial.

Not every test should be run everywhere. The most effective organizations establish test allocation strategies based on:

- The purpose of the test (e.g. logic check, real-time performance, hardware response)
- Test type (e.g. verification test, regression, fault injection, scenario simulation, acceptance test as well as requirements and compliance validation)
- Required realism and execution fidelity, purpose-driven
- Scalability and cost constraints

A few examples below:

Test goal	Best-fit platform
Check algorithm/controller stability and performance	MIL/SIL with open- and closed-loop testing using virtual prototypes
Verify integration of software modules	SIL with a focus on vECU and basic scenarios
Validate timing and IO behavior	HIL with a focus on signal-based testing
Assess complex mechanical behavior and performance	Component or subsystem test bench
Evaluate cross-domain driving functions and human interaction	DIL (driving simulator) / VIL / road tests

Table 01: Examples for test allocation

This strategic approach to mapping helps to avoid overloading high-cost platforms and ensures optimized, focused validation. Thus, a holistic testing strategy ensures that:

- Each abstraction level focuses on specific error types and behavior
- Testing is distributed, scalable, and optimized for cost versus realism
- Physical testing complements virtual validation rather than replacing it

Additionally, maintaining traceability is essential when test cases are distributed across layers. A holistic testing strategy includes smart test and (simulation) data management systems to ensure traceability, transparency and continuity across the testing pyramid to:

- Link requirements with tests and results to design or code changes
- Create easy access to and availability of virtualized development environment within the testing pyramid with high useability for non-simulation experts (e.g. calibration engineers and software developers) to leverage usage
- Enable automated reporting (ideally in compliance with regulations such as ISO 26262, etc.)

In conclusion, it is essential to establish simulation data and test management platforms that can track data and models, manage variant configurations, and ideally track test coverage and map test failures to specific system elements or software versions.

This not only allows regulatory compliance but also informed simulation and risk-based testing and continuous quality assurance.

In addition to the layered testing approach, there are testing dimensions that can be combined symbiotically. First, best practices from classical software engineering must be planned and integrated into the testing strategy. For example:

- Unit testing and static analysis which are integrated into development pipelines to detect defects during implementation
- Integration tests to identify e.g. interface, compatibility, and interaction issues within software stacks, connected ECUs or mechatronic systems
- Acceptance tests which enable the validation of e.g. control algorithms at full vehicle level, allowing for customer-centricity and integrated product quality

Secondly, it is important to differentiate between open- and closed-loop testing:

- Open-loop testing involves predefined inputs to the system under test, which are usually real measurements. These inputs do not react to the system's behavior, making it suitable for plausibility and logical checks or static evaluations (e.g. signal-level tests or communication checks in SIL or HIL). In the ADAS and AV domain, replaying raw signal data plays a crucial role for perception validation and training of AI algorithms.

- Closed-loop testing on the other hand involves direct feedback between the system under test and its environment. This is essential for validating the dynamic behavior of mechatronic control systems. Thus, closed-loop testing enables a realistic validation of control algorithms, which is particularly important for safety-critical features such as ESC and ADAS and for performance-critical differentiators like battery and range management systems.

  (Comment: Testing in SIL, HIL or on test benches can be typically performed in both open- and closed-loop and depending on the use case.)

Third, there are testing approaches, such as signal- versus scenario-based testing, which are common in the mechatronic system development domain:

- Signal-based testing focuses on verifying system behavior by stimulating individual inputs (e.g. analog/CAN signals in HIL or Simulink models in MIL) and observing the corresponding outputs, typically at ECU or component level
- Scenario-based testing evaluates and validates the overall system behavior on full vehicle level in real-world driving situations (e.g. lane changes or braking maneuvers) using simulation environments including virtual test tracks

These concepts and testing approaches can be applied and combined in different stages of the development and validation process. To combine all these testing possibilities in the best way possible, a well-thought-out and application specific testing strategy needs to be defined.

With a stronger focus on the current challenges of software-driven development, modern testing is no longer a standalone activity at different stages of the waterfall principle. Instead, it is part of an integrated, continuous development flow, a software-centric approach to vehicle development to gain efficiency. In this case, testing is naturally embedded into CI/CD pipelines and triggered automatically by software changes, model updates, and even daily commits:

- Test automation frameworks orchestrate automatic builds, validation runs, and results analysis
- Requirement traceability tools link validation artefacts to functional and safety goals
- Virtual fleets, scenario libraries, and intelligent test suites allow rapid reuse, scaling, and coverage extension

This combination enables continuous testing and deployment, not only for functions but also regarding acceptance, safety, performance, and regression testing also on system and full-vehicle level, providing high confidence in each incremental software release. The result is a continuous development approach with a continuous validation pipeline, covering everything from coding to system integration and full vehicle validation. This approach delivers confidence on a daily basis as well as at each development milestone while enabling fast iteration.

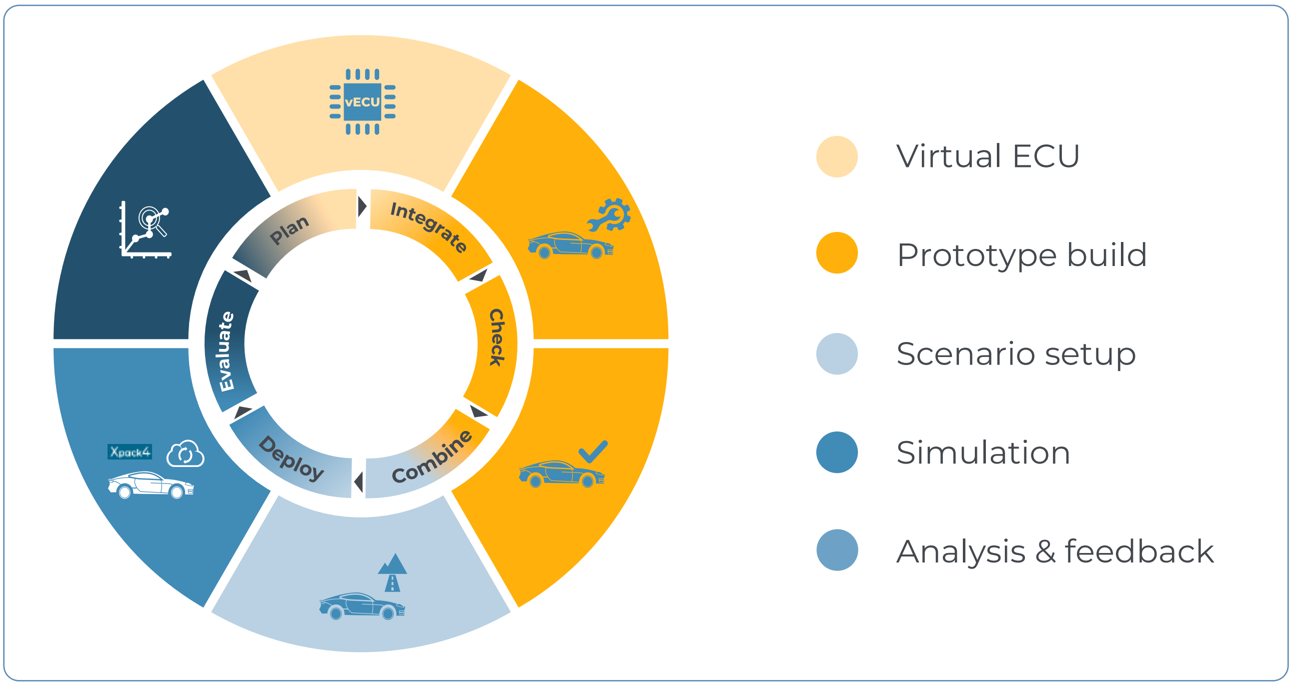


Fig. 07: Continuous development and testing on vehicle level

Together, the methods and approaches mentioned in this chapter transform testing and validation from a late-stage gatekeeper into a continuous and scalable enabler of system quality, development agility, and reduced time and costs.

Virtualized testing is not just a shortcut; it is a core pillar of future-ready automotive engineering. From this perspective, the future of testing lies in this layered, holistic, and interconnected strategy: virtual where possible, physical where necessary and always aligned with system behavior, traceability, and purpose-driven fidelity.

6. Toolchain Integration

As testing shifts to earlier stages in the lifecycle and expands across domains, it becomes clear that no single tool or platform can cover the entire validation spectrum. This has led to a major trend in the industry: seamless integration and management of tools, models, (meta)data, and processes are now a cornerstone of virtualized, efficient, and scalable testing strategies. This trend is breaking down silos and moving development departments from fragmented tools to integrated platforms.

Traditionally, development and testing tools were domain-specific and supplier-dependent, i.e. one toolset and toolchain for powertrain, another for ADAS, a third for vehicle dynamics and something completely different for infotainment and HMI. This was often linked to incompatible data formats, inconsistent traceability, and missing determinism and comparability.

Today's systems demand cross-domain coordination and continuous feedback loops. This has triggered a convergence and consolidation of:

- Simulation environments across MIL, SIL, HIL, and test benches, i.e. continuity
- Requirements and traceability tools
- Test management platforms and smart automation pipelines
- Consistent data analysis, post-processing algorithms, and results reporting systems

This enables smoother handovers, reuse of models, scenarios, and test cases as well as better collaboration between distributed teams. But it is not just about compatibility, it is about enabling end-to-end validation workflows.

7. Organizational Impact and the Human Factor: Mindset, Skill Sets, Roles and Shift in Culture

As testing becomes more virtual, continuous and integrated across domains, the impact reaches far beyond technology. It changes how teams work and collaborate with other departments, which skills are needed, and how organizations must adapt. Successful companies recognize that the shift toward modern testing and validation strategies is as much of an organizational journey as it is a technical one, since new skills, mindsets, team setups, and processes are required for the virtualized and automated validation era.

Traditional testing relied heavily on hands-on experience, deep know-how with physical components and test benches as well as real vehicle prototypes and driving skills. While this remains important, modern testing requires new skill sets as well, for example:

- Applied virtual development, modeling, and simulation expertise (deep know-how with simulation tools, different modeling approaches, scenario generation, limitations, etc.)
- Software development and test automation (CI/CD pipelines, Git and version handling systems, coding, cloud technology, etc.)
- System architecture, tool integration, and data management (integration of tools into seamless toolchains with standardized interfaces, traceability frameworks, etc.)
- Data science, optimization frameworks, and AI-based test analysis for handling large test datasets and optimization tasks

The lines between engineer, software developer, simulation expert, and tester are blurring. The main focus of validation experts is shifting from executing manual tests in the vehicle or on a test bench to designing robust, scalable test frameworks, and interpreting complex system behaviors.

From this perspective, new roles and collaboration models with cross-functional teams arise. Organizations are adapting their structure by introducing or evolving roles such as:

- Testing strategists and V&V experts/engineers, defining validation strategies across domains and abstraction levels
- Central simulation engineers, building, maintaining, managing, and validating virtual test environments and making them accessible to everyone
- DevOps/CI and cloud specialists, ensuring automated pipelines for software building, scaled, automatized testing, managing data backbones and reporting
- System integrators (incl. software and hardware), ensuring consistency and robustness across different stages and toolchains

In addition, many OEMs and suppliers are moving toward agile, feature, and customer-centric teams that own both development and testing responsibilities. This co-ownership accelerates iteration and innovation, fosters growth of skill set and improves accountability as well as mutual understanding.

As a consequence, a cultural shift must be considered: from testing as a gatekeeper and necessary hurdle to a continuous enabler. In the past, testing sometimes was perceived as a bottleneck, something that delayed release until everything was verified and validated. In the modern paradigm, testing has become a continuous enabler of progress. Especially innovators and software-centered companies from Silicon Valley have successfully proven this fact.
This shift requires:

This shift requires:

- Early involvement of testing and validation teams in the concept and design phase
- A culture of shared responsibility between development and validation
- Acceptance of a fail-fast mindset, iteration, and learning

Teams must learn to work incrementally, build testability into system designs, and treat testing as an integral, always-on activity, not just a milestone at the end.

Embracing the modern testing strategy means evolving not only tools and processes, but also mindsets and organizational models. Those who succeed will not only validate faster, but they will also have more time for innovation.

8. Implementation Patterns: How Industry Leaders Introduce and Apply Modern Testing

The shift to modern testing strategies is already underway across the automotive industry. Leading OEMs and suppliers are introducing modern testing strategies tailored to their organizational setup, product complexity, and legacy infrastructure. While no two implementations are identical, common patterns and best practices are visible.

At this point, it is important to highlight one foundation seen across all successful implementations: the central role of software-in-the-loop (SIL) and virtual ECUs (vECUs), which has become a universal validation layer across domains. It enables testing of actual compiled production code as virtual ECUs and combining them for example with virtual vehicle models.

Since virtual ECUs replicate the software behavior of a physical ECU, but can run on a classical PC or cloud instead of the target hardware, a disruption between hardware and software becomes possible.

Virtual ECUs can represent:

- Application and embedded software logic (ASW)
- Full stacks including runtime environment, operating system (OS), and basic software (BSW)
- Middleware layers (e.g. with AUTOSAR Adaptive/Classic) and virtual buses
- Containerized functions in service-oriented architectures

In this context, several levels of virtual ECUs are available, depending on the test target and required fidelity (level 0 to 4).

Virtual ECUs enable:

- Early integration and behavior testing before hardware is ready
- Rapid execution in parallel and at scale (e.g. through simulation in the cloud)
- System-level interaction testing across domains (e.g. ADAS and powertrain integration)
- Test case reuse for software from SIL to HIL to e.g. VIL

In combination with virtual vehicle models, which can simulate the mechatronic system behavior, holistic digital twins or virtual prototypes are created. They are the foundation for product- and customer-centric development on full vehicle level already in the earliest stages. Extending this framework with virtual drivers in a virtual environment, which includes static aspects like infrastructure (e.g. roads, traffic signs, etc.) and dynamic objects (like traffic, pedestrians, etc.) a digital world is created, enabling true virtual development along the full development process and the testing pyramid.

As such, SIL – from the perspective of vECUs in virtual vehicle environments – becomes the backbone of modern automotive software, function and system validation. It is fast, automatable, and scalable. In addition, SIL is capable of supporting CI/CD pipelines, requirements-based testing, and scenario-based simulation on full vehicle level already in very early development stages.

Now, having the foundation in place, common patterns at OEMs and Tier1s are:

Pattern 1: Virtual-first strategy for software-dominated domains

- **Use Cases:** ADAS & AV, HMI and UX, infotainment, vehicle connectivity, energy and range management, vehicle dynamics control systems
- **Implementation highlights:**
  I. MIL and SIL are prioritized as the primary development and validation layers
  II. vECUs used for SIL testing (open- & closed-loop, signal- & scenario-based) from day one of production code implementation
  III. Continuous integration pipelines with automated regression test suites are incorporated directly into software development workflows
  IV. HIL used mainly for real-time and communication assurance or late-stage validation (e.g. final stage integration, real-time limitations, bus communication robustness, regulatory pre-certification)
  V. Real vehicle prototypes only used for final user experience validation and subjective evaluation
- **Benefits:**
  I. Shorter development loops and earlier detection of faults in logic or interface design
  II. Rapid feedback and iteration, even after bugs occur later in the process
  III. High parallelization with low infrastructure cost
  IV. Scalable validation across variants and updates

Pattern 2: Layered testing for physics-driven, safety- and quality critical functions

- **Use Cases:** drive systems, braking, steering control, battery management, fail-operational systems like e.g. steer- or brake-by-wire, ADAS & AD systems
- **Implementation highlights:**
  I. SIL with vECUs used for early requirements validation and fault modeling
  II. Scenario-based testing is introduced early, using parameterized virtual fleets in a virtual world to expand coverage. From SIL to VIL using shared scenario libraries
  III. HIL and test benches extend validation to real-time interaction and fault injection to stress safety-critical conditions in a safe environment
  IV. End-to-end traceability for certification (ISO 26262, SOTIF)
- **Benefits:**
  I. Improved safety validation and functional coverage
  II. Early fault discovery and improved traceability from requirement to test result
  III. Streamlined compliance with ISO 26262, SOTIF (ISO 21448) and UNECE regulations to ensure high confidence in test results
  IV. Testing critical situation under safe, reproducible conditions

Pattern 3: Synergetic development of cross-domain use-cases through virtual vehicles

- **Use Case:** Full vehicle behavior involving powertrain, sensors and ADAS functions, chassis and vehicle dynamics controls
- **Implementation highlights:**
  - I. Establishment of an integration backbone by combining mechatronics models with (if required, multiple) virtual ECUs along the testing pyramid
  - II. Virtual quality gates for higher maturity levels in early stages as well as traceability and transparency (even with suppliers)
  - III. Component teams contribute validated models to a shared test architecture and immediately see their impact
  - IV. Integration and validation teams use virtual prototypes long before physical vehicles are available
  - V. Shared models, test cases and scenarios as well as KPI-based evaluation approaches, reused by software developers in SIL to test benches and proving grounds

- **Benefits:**
  - I. Cross-domain validation on full vehicle-level level possible without real hardware and prototypes
  - II. Scenario and evaluation reuse across development teams
  - III. Early system integration insights leading to reduced integration surprises and late defect discovery and a "first-time right" integration in HIL or VIL
  - IV. Foundation for digital twin-based software development loops after SOP for e.g. OTA-SW-Update validation

Pattern 4: Agile testing infrastructure for fast-changing customer expectations and SDVs in operating fleets

- **Use Case:** New E/E architectures and EV platforms as well as software-defined vehicle programs
- **Implementation highlights:**
  - I. Containerized or cloud-executed vECUs for on-demand SIL testing
  - II. Lightweight, modular HIL setups (rapid HIL farms) using standard hardware + modular I/O to reduce complexity going from full vehicle HILs to component and subsystem HILs (as small as possible)
  - III. CI/CD integration with rapid deployment and rollback capability
  - IV. On-demand simulation clusters to run extensive test matrices in parallel (in HIL and SIL)
  - V. Data-driven development: analysis, AI training, fact-based decisions

- **Benefits:**
  - I. Flexibility and scalability for fast innovation and bugfix cycles
  - II. Minimal investment for early-phase development
  - III. Enables "fail fast, learn fast" culture

Common themes across all patterns are:

- Model and test case reuse across testing pyramid to save time and costs, boost test coverage, and enable standardization
- SIL testing, also combined vECU and and virtual vehicle prototypes, as a foundation layer across all testing workflows in software-driven domains
- Scenario-based testing with scenario libraries and virtual fleets with virtual quality gates for consistency and coverage expansion
- End-2-end toolchains with centralized, consistent, and continuous data management platforms for analytics, transparency, and traceability in combination with test orchestration to automate test planning, execution, and result analysis
- User-friendly toolchains and interfaces, enabling everyone to use simulation and virtual development
- CI/CT/CD pipeline connectivity to embed testing into daily development workflows

An important general best practice for these changes is that transformation does not require a single, massive overhaul. Instead, success often comes from starting small, building modular capabilities, and scaling based on real needs and feedback.

Finally, three case snapshots on successfully evolving companies with one specific use case shall be addressed in the next chapter.

9. Case Snapshots: How Organizations Are Evolving

The shift toward true virtual development and testing test strategies is not only a technical journey, but also an organizational transformation. Below are brief case snapshots illustrating how companies across the automotive sector are reshaping their testing practices, teams and cultures. These cases, while anonymized, are based on real-world examples observed in industry projects.

Case 1: ADAS Validation Strategy Overhaul at a Global OEM

Fig. 08: Virtualized, holistic ADAS validation strategy

Challenge:

The ADAS validation team within this western OEM struggled to scale physical testing for increasingly complex Level2 / Level2++ functions. A big amount of vehicle variants, complex software integration stacks in combination with dynamic deliveries, scenario diversity, time and regulatory pressure made it impossible to rely solely on vehicle-based validation.

Transformation Path:

- Introduced scenario-based virtual validation early in the development process (starting from MIL & SIL)
- Adopted large-scaling capabilities in SIL environments and rapid HIL farms with open-loop and closed-loop testing for perception stacks and the full effect chain
- Final tuning and expert validation with subjective evaluation in VIL and with classical proving ground tests (principle: as little & as fast as possible)
- Implemented a central test data backbone across virtual and physical environments
- Created a cross-functional exchange platforms to standardize test cases and ensure reuse

Organizational Shift:

- Clearly assigned result & test ownership from “testing after development” to “testing as part of development”
- Invested in training traditional engineers in virtual development and simulation tools
- Created cross-functional teams of classical development & test engineers, software developers and simulation experts, partly also with new roles like simulation & validation strategy engineers, toolchain architects, DevOps engineers, etc.

Case 2: Scaled and Accelerated Software Testing Through CI/CT/CD

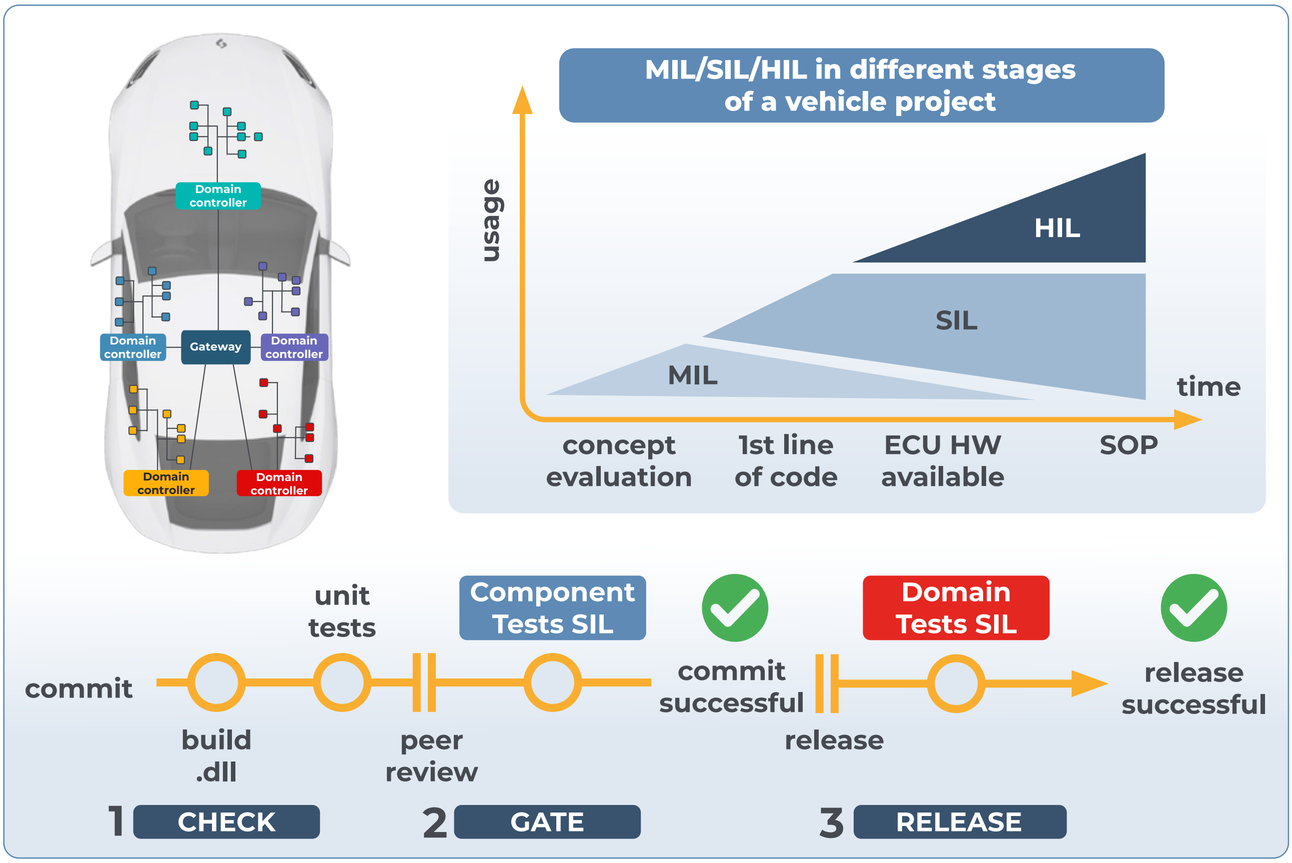


Fig. 09: Scaled and accelerated software testing through CI/CT/CD for new vehicle E/E architecture

Challenge:

A mid-sized classical OEM transitioning to EVs and new E/E architectures introduced frequent software stack updates during development, but lacked a robust strategy to test changes in a continuous, automated, and traceable way, leading to late-stage bugs, lacking transparency, traceability and unstable releases.

Transformation Path:

- Adopted a CI/CD pipeline for virtual test execution with each software build, including signal- and scenario-based testing in open- & closed-loop-frameworks with vertical integration tests (unit-, integration-, acceptance-tests)
- Created a test automation framework covering MIL, with a focus SIL and finally integrating HIL for real-time conditions and BUS communication tests
- Standardized test case catalogues, regression tests & priorities with automated, KPI-based result analysis

Organizational Shift:

- Established „Virtual Quality Gates“ for virtual prototypes and virtual ECUs
- Integrated testing teams and simulation experts with software development
- Shifted team KPIs from “number of tests” to strategic and optimized test coverage

Case 3: Holistic Vehicle Dynamics Control Development and Testing

Fig. 10: Virtualized vehicle dynamics control development and testing

Challenge:

A market leading, globally acting Tier1 aimed to optimize ESC and brake control strategies through simulation but lacked clarity on where to test what and how to get valid virtual prototypes. Moreover, it was critical to keep consistency from SIL to HIL and road tests to set the frame for virtual releases and homologation.

Transformation Path:

- Introduced objective, KPI-driven development targets for function and system behavior
- Incorporated virtual ECUs and full-vehicle models for early-phase testing until function release, the OEM had to deliver valid virtual prototypes (passive) ready for virtual homologation
- Standardized maneuver catalogues for consistency, that span from the use cases of function development and parameter tuning in simulation to physical testing and validation
- Continually improving model correlation along development process in combination with simulation credibility measures and combined open- (i.e. replay) and closed-loop tests
- Introduction of Sensitivity Analysis and Optimization Frameworks for virtual parameter variation, ECU calibration and test strategy (starting in MIL & SIL)
- Developed traceable, automated test strategies for continuous delivery

Organizational Shift:

- Formalized “Virtual-First” culture within the organization and their target customers, including building blocks for successful virtual development
- Unified simulation and physical test teams under a shared responsibility model, creating interdisciplinary virtualization project teams
- Appointed a Test Strategy Lead to orchestrate platform usage, timing and test coverage

These case snapshots demonstrate that technical enablers such as MIL/SIL/HIL in combination with digital twins in a virtual environment are only part of the picture. Real transformation happens when organizations adapt their structures, roles, and decision-making approaches to match the complexity of modern development requirements. The companies that succeed are not just developing and testing differently, they are following new paths, starting from their current organizational boundary conditions and setup.

10. Summary and Strategic Recommendations

Automotive development is undergoing one of the biggest transformations in its history: greater complexity, software-driven development, and radically shorter innovation cycles. Holistic testing strategies must therefore evolve in parallel, to enable the necessary transition, from a late-stage validation gate to a virtualized, continuous, layered, and integrated quality enabler across the vehicle lifecycle.

This white paper has shown that leading organizations are adopting modern, virtualized testing strategies that combine early-stage virtual validation, virtual ECUs and SIL, modular HILs, automated pipelines, and smart test allocation. These strategies are not only technologically feasible, but they are also becoming a competitive necessity, increasing development efficiency and ensuring safety, quality, and innovation at speed.

Key takeaways are:

- SIL and virtual ECUs are foundational for fast and integrated testing, not optional. From early-stage validation to cross-domain integration and safety assurance, SIL and vECUs are the bridge. They reduce reliance on hardware, shorten timelines, and enable scalable, automation-ready, holistic test strategies across the development lifecycle. Whether for verifying logic and controls, checking integration or replaying real-world scenarios, SIL acts as a bridge between early model-based testing and late-stage real-time execution. As a result, it is increasingly reused and combined with HILs and test benches as well.
- Tests do not have to be executed everywhere. A holistic, smart test strategy allocates the test goals to the right platform and complexity level (purpose-driven fidelity), minimizing time and costs.
- Traceability and automation are essential to make the pyramid scalable and enable software-driven development.

To support transformation, the following strategic actions and recommendations are relevant:

1. Holistic test strategy from day one

- Requirements, systems, functions, models, ECUs, and hardware must be mapped to appropriate test platforms, with timing and purpose clearly defined: shift-left and virtual-first principles are essential
- Reliance on hardware should be minimized by scaling virtual and early testing
- Emerging simulation technologies need to be monitored and integrated into existing processes
- A robust SIL environment with vECUs and virtual prototypes (ideally fleets and scenarios) must serve as a scalable foundation
- Testing should be optimized using best practices from software, mechatronics, and data-driven approaches (signal vs. scenario, open- vs. closed-loop, testing pyramid alignment)

2. Matching test type to platform

- Test platforms must be selected based on purpose and required realism, not legacy or convenience
- SIL is suited for logic and software integration; HIL for real-time conditions and BUS communication; test benches and VIL for complex, physical system behavior
- In domains such as ADAS and powertrain, simulation should precede hardware availability to validate software
- Software and Hardware integration must be staged to manage complexity
- Speed, realism, and coverage must be balanced in a documented, traceable strategy
- Over-testing on expensive or late-stage platforms should be avoided through early virtualization

3. Tool integration and automation across layers

- Test orchestration pipelines must link design, software builds, and test results to enable continuous delivery
- Execution, reporting, and traceability should be automated
- Test and maneuver catalogs, along with scenario libraries, must be used to manage and allocate test cases consistently
- Test case reuse across SIL → HIL → test bench → VIL must be ensured
- CI/CD processes must include simulation-based test and release stages
- AI algorithms should be applied to accelerate tasks such as documentation and data management

4. Cross-disciplinary skills and roles

- Engineers, developers, and testers must be upskilled in virtualization, simulation, automation, scripting, and system integration
- Roles such as system architects, validation strategy experts, and DevOps engineers must be established or supported
- Interdisciplinary collaboration must be promoted across previously siloed teams
- Where feasible, customer-centric, agile and cross-functional teams should be formed

5. Modular and small start, intentional scaling

- Full transformation in a single step must be avoided
- Pilot projects in high-impact areas (e.g. SIL-based validation for ADAS or drive systems) should be prioritized
- KPIs and ROI metrics (e.g. bug detection rates, release speed, traceable coverage, effort savings) must be tracked to support strategic decisions, especially for cloud and scaling use cases

6. Traceability is nonnegotiable

- Systems must be implemented that link requirements, test cases, and results across abstraction levels to enable consistent systems engineering
- Traceability data must support risk-based testing and continuous validation, contributing to simulation credibility
- Safety and regulatory readiness (e.g. ISO 26262, SOTIF) must be ensured through structured traceability
- Manual or spreadsheet-driven approaches are insufficient; automation of traceability processes is essential

11. Closing Thought

Testing transformation is not just a technical upgrade, it is a strategic shift in the way automotive products are built and validated. In software-driven, interconnected domains like ADAS and AV, powertrain, and vehicle dynamics, validation challenges cannot be mastered by hardware-centric or data-driven testing alone. A holistic approach, combining the latest, proven testing and validation technologies from different domains, is the key to solving the validation challenges under optimized time and cost measures.

Those who invest early in virtual validation, automation, and holistic test strategies will not only deliver better products faster, but also create the organizational agility to thrive and innovate in an increasingly software-driven and interconnected world.

Solutions for Virtual Vehicle Development

At IPG Automotive, we provide cutting-edge simulation software, hardware solutions and consulting services to enable realistic testing and validation of vehicles in a virtual environment.

Our solutions support every stage of the development cycle: They ensure a continuous process based on a consistent virtual vehicle prototype as well as uniform evaluation criteria and tests throughout development.

Do you still have any questions? We are happy to answer them.

Germany
IPG Automotive GmbH
Fautenbruchstraße 46
76137 Karlsruhe

France
IPG Automotive France SARL
Vélizy Espace
13, avenue Morane Saulnier
Immeuble Nungesser
78 140 Vélizy-Villacoublay

United Kingdom
IPG Automotive UK Ltd.
Blythe Valley Innovation Centre
Central Blvd
Solihull
B90 8AJ

Sweden
IPG Automotive Sweden AB
Bergfotsgatan 4
SE-431 37 Mölndal

USA
IPG Automotive USA, Inc.
4110 Varsity Drive
Ann Arbor, MI 48108

India
IPG Automotive India Private Limited
Unit#3F-03, Level 3, Century Central
Mango Garden Layout
Bikasipura
Kanakapura Main Road
Bangalore – 560062

Japan
IPG Automotive K.K.
9th Floor, ARK Hills South Tower
1-4-5 Roppongi, Minato-ku
Tokyo 106-0032

Korea
IPG Automotive Korea Ltd.
C-1013, 128,
Beobwon-ro, Songpa-gu
Seoul, 05854

China
IPG Automotive (Shanghai), Ltd.
Room 1203 & 1205
China Overseas Center Tower A,
No.2, Lane 699, Tongchuan Road
Putuo District
Shanghai, 200333

www.ipg-automotive.com
web-contact@ipg-automotive.com