

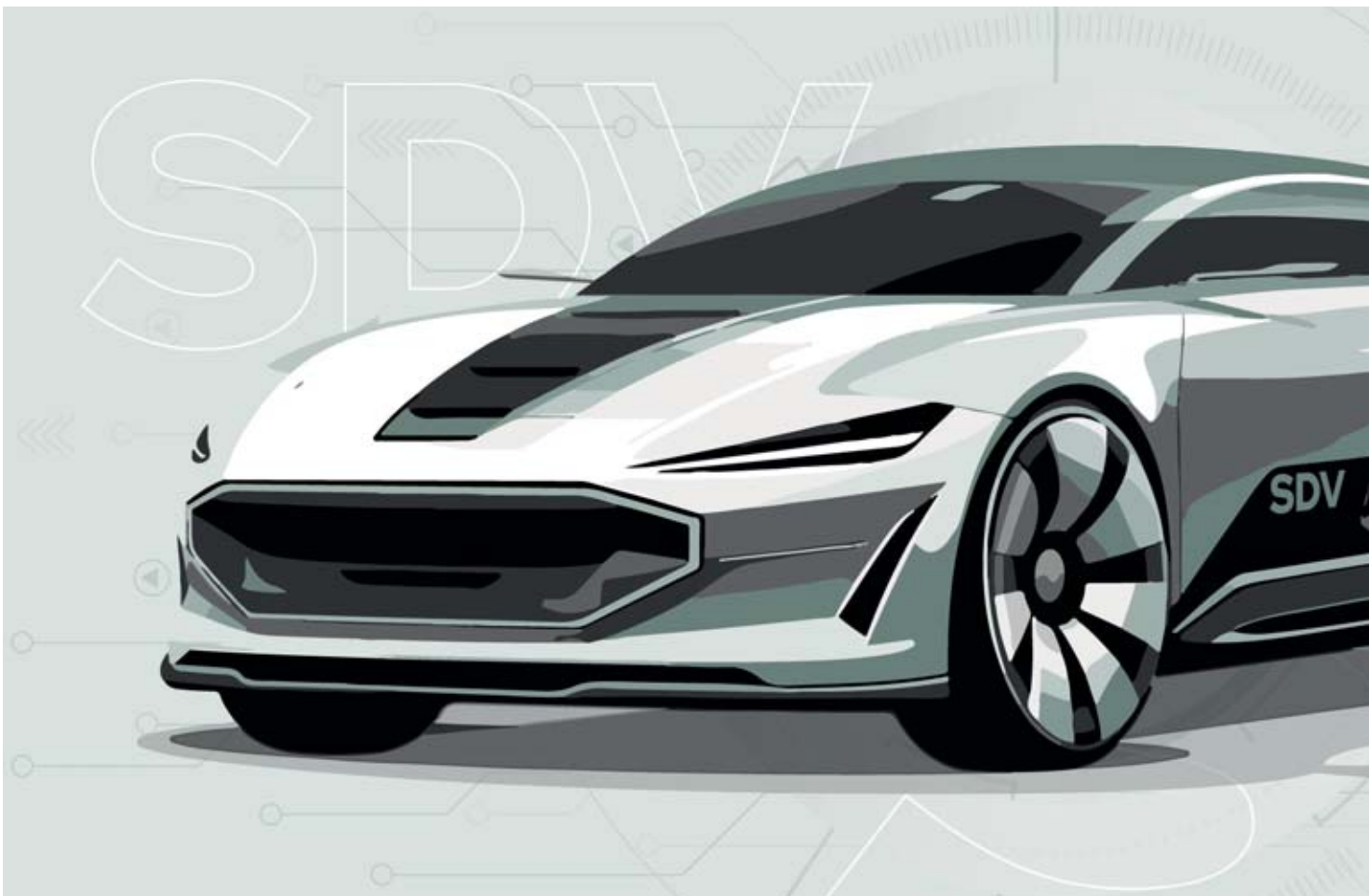
## Integration debt in the SDV

# AI and orchestration accelerate the implementation of SDVs

Fragmented tools and data silos slow down the development of new SDVs. Cloud-native platforms can help: They connect distributed organizations, reduce staffing requirements, cut costs, and enable end-to-end integration and validation processes based on transparent, data-driven decision-making.

The shift toward the software-defined vehicle is putting immense pressure on traditional integration and validation processes. New features are being developed in shorter iteration cycles, the scope of testing is expanding, and updates are being distributed at shorter intervals via over-the-air updates. Meanwhile, the requirements for functional safety, cybersecurity, and homologation remain as high as ever. For OEMs, this creates a clear economic and technical conflict of objectives: integration costs must be reduced significantly to remain competitive without compromising quality, safety, or user experience due to the growing share of software. Therefore, the key is not to perform more tests, but

to design integration and release processes that scale efficiently and remain resilient under high-frequency workloads. In practice, it becomes clear that the problem is not the speed of individual teams, but the transitions between organisations, domains, integration levels and test environments. Gaps, data silos, and manual coordination loops arise here, costing time and prolonging risks. Hybrid-capable platforms such as one:cx address this bottleneck by serving as a release control board for integration and validation. These platforms connect existing tools, create a central data hub, and orchestrate the entire integration and validation process across all test environments, as well as across integration levels, from units and components to subsystems and the



overall system. This reduces integration costs while ensuring quality and a positive user experience through reproducible releases.

### The integration problem in the SDV era

In SDV projects, a contradiction often arises: individual teams appear efficient and deliver quickly locally, yet the overall flow stalls. The reason lies in the transitions between integration stages. What appears stable at the unit level behaves differently at the component level. What works at the component level reveals new conflicts at the subsystem level. At the system level, effects emerge that were previously invisible or could not be clearly attributed. Reasons for this include different toolchains, separate data storage, differing configurations, and distributed roles and responsibilities, among others. This chain of problems persists as integration increases, costing time and jeopardizing the SOP. The cause is structural: each test environment operates as an isolated silo with its own infrastructure and data storage. Results must be collected, consolidated, and evaluated manually, a process that becomes nearly unmanageable as the scope of software in the SDV grows exponentially. This gives rise to integration debt, which accumulates at the interfaces between integration levels, teams, domains, and system environments. In other words, it

accumulates wherever results are not consistently traceable and software quality can only be determined by manually aggregating distributed data.

### From sequential integration to continuous processes

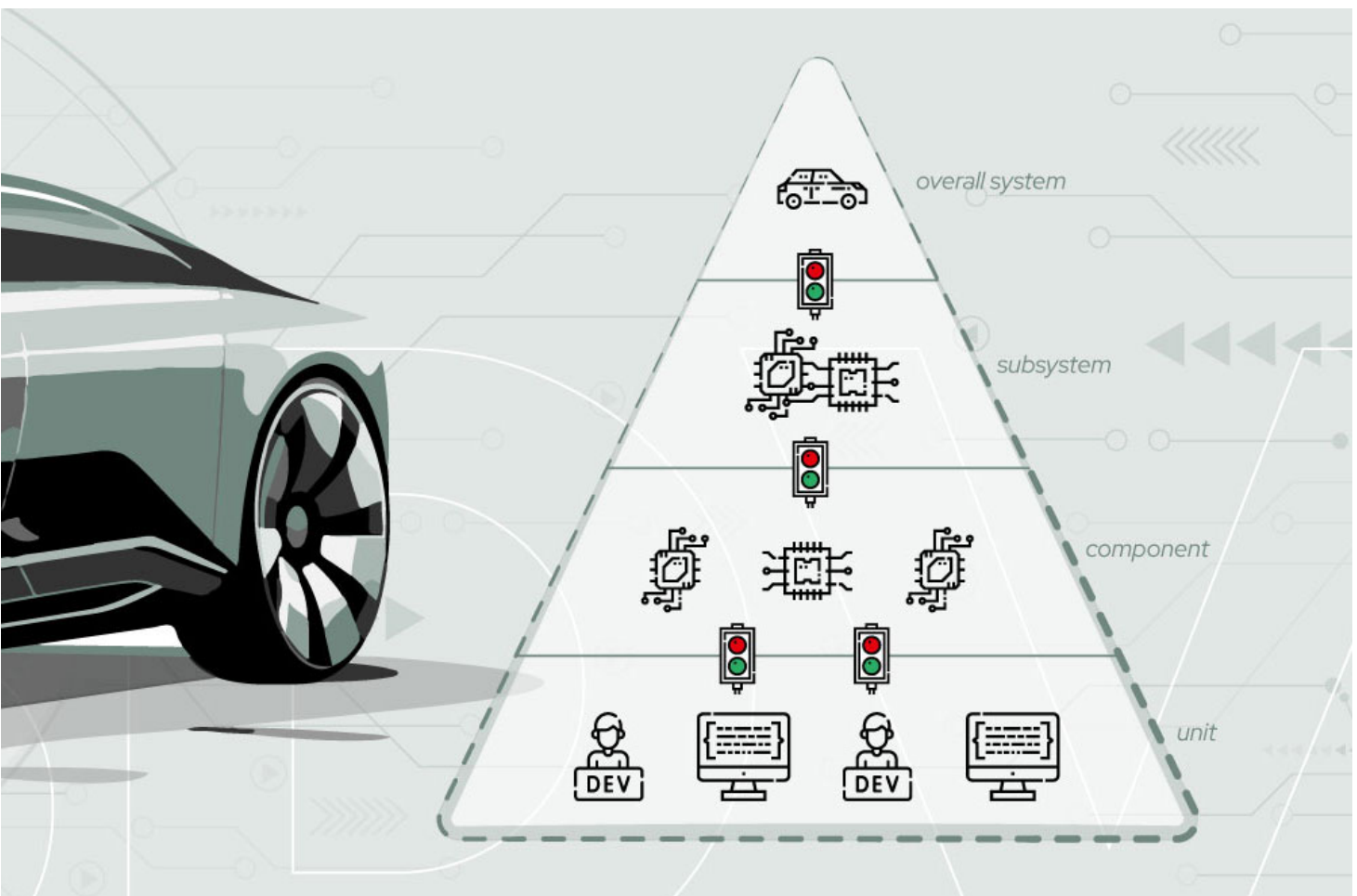
The classic V-model is evolving with the addition of continuous testing (CT) and continuous integration (CI). Despite shift-left approaches, however, integration challenges persist. Consequences include defects being detected too late, high-risk releases leading to costly recalls, and a high level of manual effort. To address the increasing complexity and frequency of SDV development, a strategic realignment with continuous quality assurance is required.

**Overcoming silos:** End-to-end orchestration and centralized data management reduce friction between heterogeneous environments (SiL, HiL, and ViL) and domains.

**Cost-effectiveness through end-to-end integration:** CI/CD pipelines automate the testing process. AI agents assist with test specifications, test cases, and analyses, reducing manual effort.

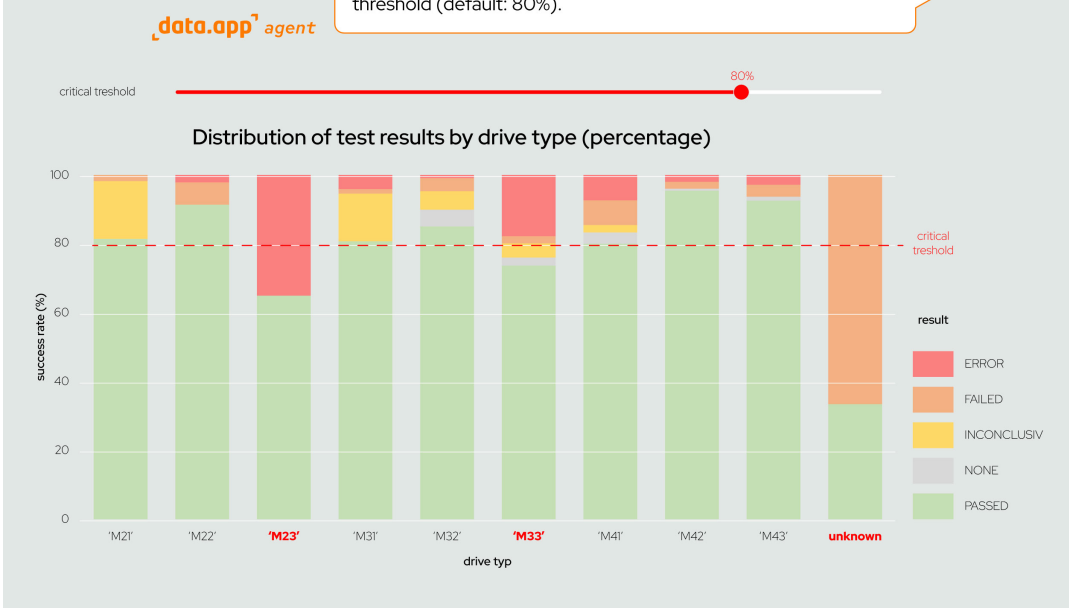
**Efficient resource utilization:** The organization's global testing resources are automatically coordinated and utilized around the clock to optimize usage, especially during nighttime hours.

How do AI and orchestration overcome integration debt in SDV development and accelerate secure releases?



I need all the test results broken down by drive type. Display them as stacked bar charts with percentages and sort them alphabetically. The critical threshold should be adjustable, with 80% as the default setting.

I am creating a stacked bar chart showing the test results as percentages by drive type, sorted alphabetically, with an adjustable threshold (default: 80%).



An AI-powered agent assists with analyzing and communicating status by generating context-aware views and visualizations using natural language based on existing evidence.

**An orchestration platform is preferable to a collection of tools.**

Many organizations already have a powerful ecosystem consisting of CI, management systems, test frameworks, HiL landscapes, and customer-specific tooling. However, what is missing is an overarching layer that connects and controls the involved systems. This is exactly where one:cx comes in: an enterprise platform that integrates and monitors tools, infrastructures, and workflows across the organization, consolidates results, and enables data-driven release decisions.

**Key features of one:cx**

- **Process automation:** Automate processes and workflows with quality gates and flows to ensure a seamless CI/CD pipeline.
- **Single source of truth:** A central data hub eliminates silos and provides consistent information, including test results and logs from SiL, HiL, and vehicle scenarios; configurations; and requirements from ALM systems. The platform thus creates a trustworthy data foundation for all stakeholders.

such as software repositories, requirements tools, ticketing systems, XiL platforms, as well as customer-specific test frameworks and test benches.

- **Intelligent test planning and execution:** Thanks to comprehensive test orchestration and intelligent scheduling, tests are automatically distributed to suitable XiL systems, engine test benches, or even the fleet, and executed in parallel.
- **AI agents:** AI services reduce bottlenecks in evaluation and verification. Based on automated workflows and consistent data in one:cx, AI agents assist in deriving test specifications and test cases, cluster and prioritize failure patterns across releases and variants, and detect anomalies as well as unstable tests. These efforts significantly reduce review and analysis time while ensuring that approvals are deterministically validated through quality gates and traceability.

**Accelerated approvals**

Artificial intelligence (AI) demonstrates its added value in the SDV validation process precisely where traditional automation reaches its limits. Automation provides consistent, predictable results when the process and input data are clearly defined. However, as the SDV cycle accelerates, the bottleneck shifts from execution to evaluation. Thousands of results must be categorized, summarized, and translated into a release decision. AI complements the process here without replacing the deterministic nature of quality gates.

**Quality gates automate approvals.**

While AI-powered analyses primarily reduce the manual effort required to evaluate individual test results, quality gates address the decision-making level. In the CI/CD pipeline, approvals are based on clearly defined and automatically verified criteria, rather than being made ad hoc or manually. With these rule-based approval decisions, only builds of sufficient quality proceed to the next stage, such as integration, system, or road testing. The core concept is quality gate plans. These plans describe the approval process across

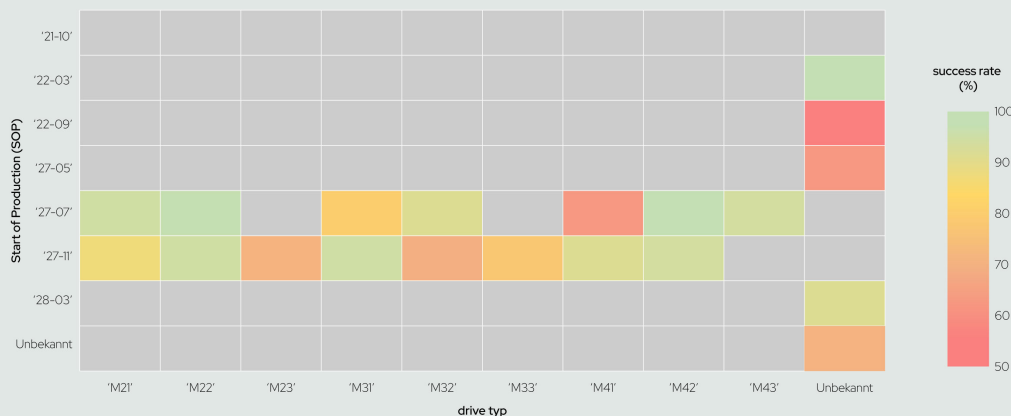
- **Cross-domain traceability:** Standards such as ISO 26262 or ASPICE require seamless traceability. With one:cx, every test result is uniquely assigned to a scenario, software version, vehicle variant, or the relevant requirements. Every change at every point in the process is automatically documented, providing direct support during debugging.
- **Integration of existing tool landscapes:** one:cx is designed for seamless integration and uses standardized interfaces. Rather than replacing existing tools and solutions, one:cx connects and centrally orchestrates them, including systems

Also, show me the relationship to the Start of Production (SOP). To do this, create a heat map: test results by drive type and SOP.

I am adding a heat map to the visualization that shows the relationship between drive type and SOP (Start of Production).

data.app agent

Heatmap: Test results by drive type and SOP



For example, test results can be broken down by drive type, dependencies can be displayed as a heatmap, and anomalies can be analyzed in more detail, all without requiring specialized knowledge of dashboards or BI tools

multiple stages and define the conditions that a build must meet in each phase. The platform automatically checks these conditions and visualizes the current status.

If a gate is blocked because a criterion has not been met, this status becomes visible immediately and propagates to subsequent stages. Further testing does not occur until the preceding stage confirms the required quality. This ensures that testing resources are used efficiently and that problems are identified early, before they escalate to the subsystem or system level. Quality gates are not rigid control mechanisms; rather, they address the dynamic requirement of ensuring the necessary maturity of the build at all times without blocking all further steps in areas that require attention, whether known or planned.

#### AI consolidates evidence and focuses reviews.

As test automation increases, the bottleneck shifts to the evaluation phase. The scarcity lies not in the execution itself, but in the time available to experts to review, categorize, and prioritize thousands of results. AI and machine learning services handle the initial, data-driven consolidation and provide a robust pre-structure for the review decision.

- Failed tests are compared to historical failure patterns across releases, variants, and integration stages, then grouped into patterns.
- Logs and signals are analyzed for anomalies and recurring error patterns.
- Correlations between errors and environmental factors such as configuration, load profile, or time window are highlighted.
- Unstable tests are identified and handled separately so they do not skew release decisions.

This results in a shift from a blanket review to targeted prioritization.

Rather than starting with a long list of failed tests, reviews now begin with a few clearly defined problem clusters that are highly relevant to the release. Aggregated error patterns and their relationships are displayed. Teams can quickly identify dominant clusters, their relationships, and worthwhile areas of analysis. This reduces unnecessary repetitions, makes root cause analyses more precise, and prepares releases faster because the necessary evidence is already available in a structured, traceable format. Although the final decision remains with humans, it is significantly facilitated by clear patterns, priorities, and context.

#### Dashboards and KPIs

Many development organizations lack a unified and reliable view of their current integration and quality status. Questions about this status are answered through data exports, manual evaluations, and meetings. This process is time-consuming and leaves room for interpretation. In the SDV cycle, the analysis is often outdated by the time it is complete because hundreds of commits, builds, and test results are added daily.

The hybrid-capable platform one:cx creates a shared, always-up-to-date data foundation that is consolidated not by tool or team, but along the release flow.

Role-based, real-time dashboards make the status measurable and comparable. Additionally, an AI-powered agent assists with analyzing and communicating the status. Rather than creating new reporting silos, it generates context-aware views and visualizations using natural language based on existing data. This transforms data availability into operational decision-making capability with consistent key performance indicators (KPIs) for engineering and a robust management view of maturity, risks, and release readiness. (na/bs) ■ Author: Jan Georges, Business Development Solution Lead at traceTronic