WHITEPAPER

Generative AI in Android IVI Programs

Traceable Engineering Framework for ASPICE Compliance



Debashis Panigrahi
Senior Solutions Architect, Sasken Technologies Ltd

November 5, 2026

SASKEN

Table of Contents

01 Abstract

02 Introduction: The Problem We're Solving
- The New Reality of IVI Development
- The Core Friction Point
- The Cost of Not Solving This

03 The State of Play: Background and Related Work
- The ASPICE Imperative
- What OEMs Should Expect from Suppliers

04 Our Approach: The Prompt and Output Evaluation Workflow (PEW)
- The Core Principle
- The Five-Stage Workflow
- The Evidence Bundle
- What PEW Is Not

05 How It Maps to ASPICE
- SYS.2-3: System Requirements and Architecture
- SWE.1-2: Software Requirements and Architecture
- SWE.3: Implementation and Review
- SWE.4-6: Verification and Testing
- The Two-Column Discipline

06 Results from Our Internal Reference Program
- What We Measured
- Hygiene Signals
- What This Data Represents
- How Reviewers Actually Use This

07 Conclusion: What This Means for You
- The Bottom Line
- Recommendations by Role
- Five Questions for Your Next Supplier Review
- Limitations and Caveats
- Next Steps
- A Final Note

08 Selected references

09 About the Author

10 About Sasken

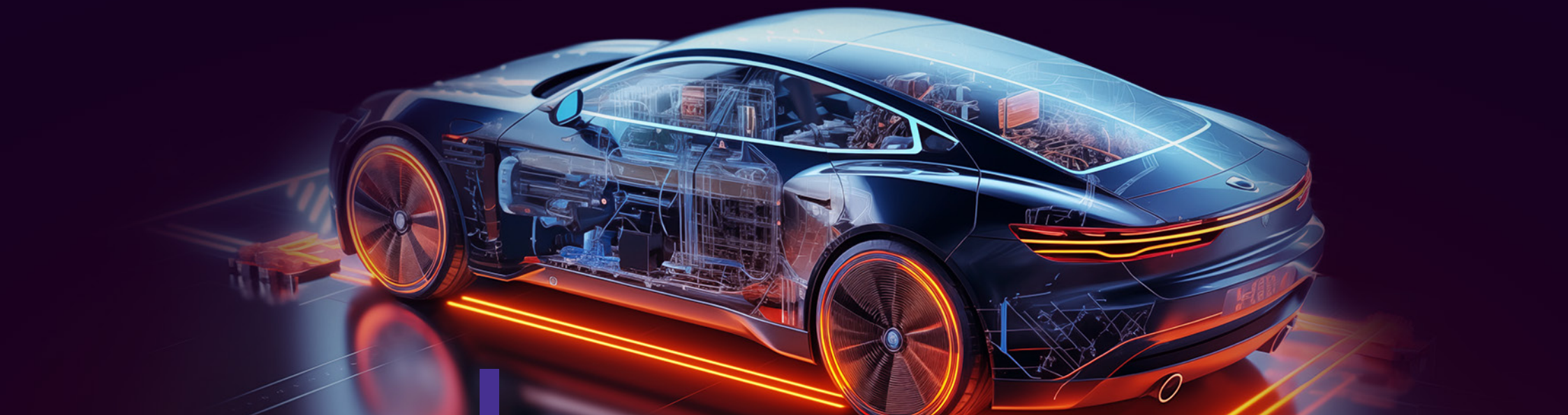
Abstract (Note: Covers brief of the White paper)

The automotive industry is at an inflection point. Generative AI tools are becoming ubiquitous in software development, 61.8% of developers now use AI assistants and 76% either use or plan to use them in their daily work. For Android IVI programs, these tools promise dramatic productivity gains, with studies showing 55.8% faster task completion and 21% time reduction in controlled enterprise trials.

But here's the friction point: while AI dramatically accelerates code generation, the expectations around traceability, verification, and auditability haven't shrunk. OEMs still demand the same rigorous evidence for every change: Which requirement does this address? Who reviewed it? What tests or analysis cover this build?

This whitepaper addresses that gap. We propose Prompt and Output Evaluation (PEW) Workflow, a tool-agnostic discipline that ensures every promoted change carries a complete evidence bundle: work-item reference, independent review, machine evidence (CI, test results, static analysis) and named approval. PEW provides traceability evidence that supports Automotive SPICE (ASPICE) work products and assessment activities (particularly SYS.2-3 and SWE.4-6), providing repeatable anchors for prompts and evaluations.

PEW is designed as a practical checklist that OEM and Tier-1 reviewers can use immediately, not as a replacement for a full ASPICE assessment, but as a daily operating rhythm that makes GenAI-assisted development auditable at merge time.



Introduction: The Problem We're Solving (Problem Statement)

1. The New Reality of IVI Development

Software-defined cockpits have transformed automotive development. Today's Android Automotive IVI systems integrate multiple apps such as media, navigation, telephony, projection and all sharing complex intents and lifecycle contracts. A seemingly minor launcher change can cascade into bench failures across the entire system.

Now overlay generative AI on this complexity. Development velocity is accelerating, but the fundamental expectations around engineering discipline haven't changed. OEM review gates still demand traceability, evidence, and accountability for every merged change.

2. The Core Friction Point

Here's the reality: AI tools increase the volume of diffs, but review obligations don't shrink because of it. Chat logs are not review records. IDE sessions are not audit trails. And "the AI generated it" doesn't answer the question of why a change was made or how it was verified.

Suppliers are gaining drafting speed, yet many still can't produce a single auditable packet per merge when an OEM challenge arrives. The result? Integration drag, rework, and finger-pointing that erode the very productivity gains AI promised.

3. The Cost of Not Solving This

When teams push fast merges without bundling evidence upfront, they consistently hit integration drag once work lands on the bench. OEMs spend countless hours in reconstruction meetings, trying to piece together what happened. Tier-1 suppliers face delayed approvals and quality escapes that damage trust.

The PEW addresses all this by making the evidence bundle a prerequisite for promotion not an afterthought.



The State of Play: Background and Related Work

1. The ASPICE Imperative

ASPICE remains the shared language for work products and traceability in the industry. GenAI changes the volume of material pushed into review, not the review obligation itself.

Every ASPICE process area still requires evidence:

a. **SYS.2-3:** System requirements and architecture traceability.
b. **SWE.1-2:** Software requirements and architecture.
c. **SWE.3:** Implementation and review.
d. **SWE.4-6:** Verification and testing.

PEW sits on top of these as the daily operating rule that ties every merge to ASPICE evidence requirements.

2. What OEMs Should Expect from Suppliers

Before trusting a supplier with GenAI-assisted IVI delivery, OEMs should see four foundations in place:

1. **CI/CD and governed assistants** implemented and deployed, with clear role checks under tool allowlists
2. **Traceability** from requirements through architecture to implementation, terminating cleanly in the ALM (not in side channels)
3. **An internal reference platform** to validate integration ahead of customer milestones
4. **A bounded incubation:** typically ninety days on a single vertical, before rolling out program-wide

PEW addresses the governance gap between these foundations and daily development operations.



Our Approach: The Prompt and Output Evaluation Workflow (PEW)

1. The Core Principle

At its heart, PEW rests on one rule:
Humans stay in charge of architecture decisions, release management, and qualification. Using an assistant is a label on the record, not a waiver from any obligation.

Every change that gets promoted must state:

- **What** was changed
- **Why** it was changed
- **What evidence** proves it works
- **Who** approved it

2. The Five-Stage Workflow

PEW breaks down into five sequential stages for each merge:

Stage 1: Intent and Scope Definition

Before any code is generated or modified, the work item is defined with:

- Requirement references
- Modules touched
- Explicit non-goals
- Any safety or security flags requiring attention

This sets the boundaries for what the assistant can do. If a supplier cannot fill out this scope before writing a single line, the work isn't under control.

Stage 2: Constrained Generation

AI-assisted output is produced using:

- Versioned playbooks
- Enforced coding standards
- An allowlist of approved tools

Changes are kept to merge-sized granularity, no sprawling diffs that are hard to review.

Stage 3: Technical Review (Review A)

A reviewer checks the output for correctness against:

- Stated requirements
- Broader IVI architecture

This is where design and implementation get challenged.

Stage 4: Process Review (Review B)

The same change is examined for:

- Traceability | Test coverage | Documentation updates | Proper recording

Critical rule: Organizations should define reviewer independence criteria appropriate to program criticality, ensuring reviewers can objectively assess generated outputs and associated evidence.

Stage 5: Verification Package and Promotion

All evidence is assembled:

- Static analysis outputs
- Build identity
- CI logs
- Applicable test results

Only then does a named approver sign off. Only when the package is complete does the change get promoted.

3. The Evidence Bundle

PEW transforms informal outputs into concrete items before promotion:

What Was Informal	Becomes Formal Evidence
Chat session	ALM work item IDs
“I tested it”	CI execution records
“Looks good to me”	Formal review records with named reviewer
“Lint was clean”	Static analysis exports
Assistant output	Traceable back to prompt and scope

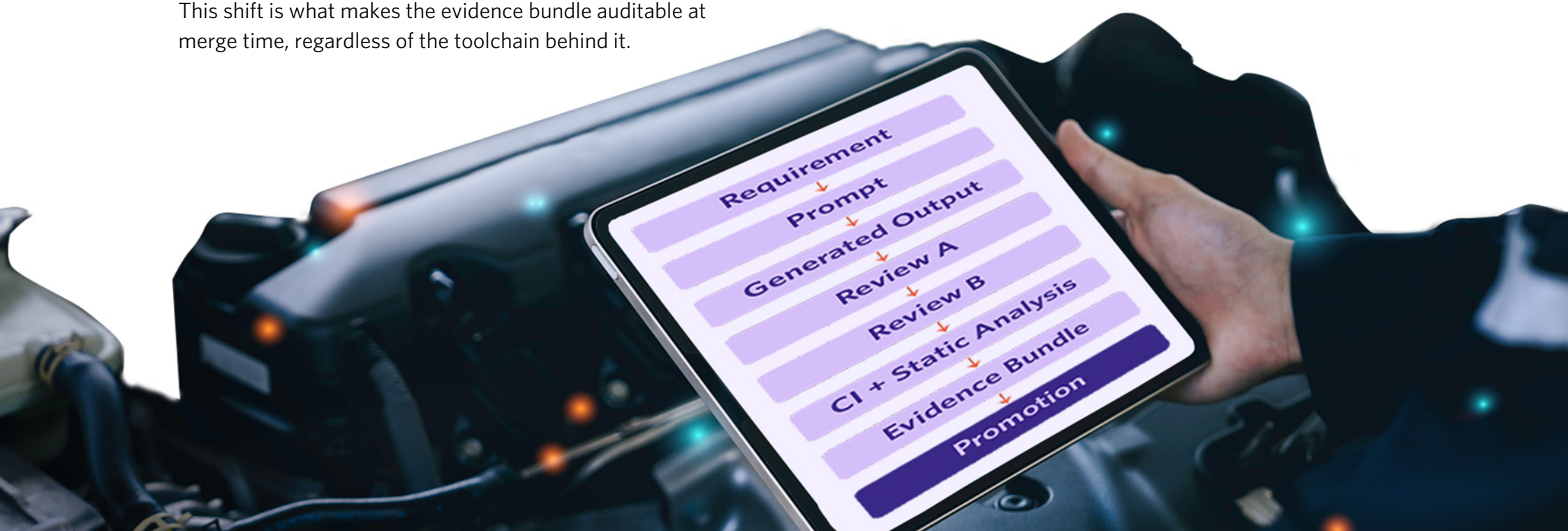
This shift is what makes the evidence bundle auditable at merge time, regardless of the toolchain behind it.

4. What PEW Is Not

PEW is not:

- A full ASPICE assessment methodology
- A silver bullet for all quality or safety concerns
- A replacement for human judgment and expertise
- A claim of ISO 26262 or ISO/SAE 21434 compliance

It is a practical daily discipline that makes GenAI-assisted development traceable and auditable.



How It Maps to ASPICE

PEW provides concrete anchors for ASPICE process areas. Here's how to use this at the gate:

1. SYS.2-3: System Requirements and Architecture

Supplier provides (before writing code):

- Baseline or release identifier
- Affected requirements
- Relevant interfaces
- Explicit non-goals

Reviewer verifies (at merge):

- Delivered output matches baselined system requirements
- Architecture alignment is maintained
- No silent expansion of scope occurred

2. SWE.1-2: Software Requirements and Architecture

Supplier provides:

- Software requirement IDs
- Architectural constraints that apply

Reviewer verifies:

- Upward traceability to requirements
- IVI coupling risks identified during review are closed off

3. Implementation and Review

Supplier provides:

- Modules touched
- Version of coding standard in use
- Merge-request size limit compliance

Reviewer verifies:

- Independent review took place (Review B)
- Static analysis ran and passed on affected paths
- CI executed correctly on those paths

4. SWE.4-6: Verification and Testing

Supplier provides:

- Test case IDs (or coverage scope if not case-based)

Reviewer verifies:

- Adequacy has been declared
- Gaps are explicitly called out
- Release checklist is satisfied before sign-off

5. The Two-Column Discipline

This two-column approach gives a gate reviewer a concrete, repeatable way to ask the same questions regardless of whether AI was used. It keeps the evidence bundle tied directly to ASPICE process areas and prevents AI assistance from creating traceability gaps.

Results from Our Internal Reference Program

1. What We Measured

We applied PEW to an internal AAOS-style multi-app IVI reference program. The scope included: Launcher, Media, Bluetooth, Navigation, Projection, Speech, Shared services

2. Hygiene Signals

Measure	Value	How to Read It
Static analysis (module A)	0 errors, 81 warnings	Implementation hygiene snapshot [8]
Static analysis (module B)	0 errors, 68 warnings	Same; not release sign-off
Warnings (combined)	149 total	Trend target for remediation, not a gate pass/fail
Cold-start samples	~822 / 832 / 788 / 786 ms	Single-context observation; not a benchmark

Percentage merges with complete evidence bundle for ASPICE traceability also reduced.

3. What This Data Represents

We are not claiming a controlled before-after study here. The real operating value of PEW isn't a performance number, it's methodological.

When CI logs, static analysis outputs, and review records are consistently attached to every merge, the OEM's questions shift from trying to reconstruct what happened to simply making a judgment about what's in front of them.

4. How Reviewers Actually Use This

In practice, reviewers are less concerned with which particular assistant was used and more focused on whether:

- The change is properly bounded
- It has been reviewed independently
- It can be reproduced from stored artifacts

PEW takes informal outputs, chat sessions, ad-hoc testing, verbal approvals and turns them into concrete, auditable items before promotion.



Conclusion: What This Means for You

1. The Bottom Line

When GenAI is used in IVI development, the real payoff comes when every meaningful merge tells the same evidence story you already expect from any other change. PEW makes that repeatable by providing ASPICE-aligned anchors and a single promotion gate that applies equally to all work.

2. Recommendations by Role

For Tier-1 Program and IVI Leaders:

- Run a 90-day PEW pilot on one vertical
- Block shared-branch promotion unless bundle fields are fully populated
- Track metrics like review rework and time to green CI

For OEM Engineering and Sourcing Teams:

- Use the ASPICE-aligned prompts in Section 5 as a script for supplier reviews
- Insist on environment parity: branch, job identifier, target, and reviewer independence must be recorded

For Platform Owners:

- Ensure CI/CD, traceability, and a reference platform are in place before scaling AI assistants across the program

3. Five Questions for Your Next Supplier Review

1. Show one recent merge: What is the requirement or defect ID? Who was the reviewer? What proof exists that the reviewer did not run the assistant session themselves?
2. Where is the CI output for that commit? Which modules were covered by static analysis?
3. For a change under SYS.4-5: What is the environment ID, the build under test, and the named test with its pass/fail result?
4. What was explicitly out of scope for the last AI-assisted task, specifically, the non-goals recorded in the prompt?
5. What is your 90-day plan to reach 100% bundle compliance on a single IVI vertical?

4. Limitations and Caveats

We should be clear about what this whitepaper doesn't claim:

- The metrics presented are internal snapshots
- External studies are directional only
- Formal outcomes for ASPICE or cybersecurity or R155 require program-specific evidence to be assessed

PEW and Cybersecurity Traceability is important to cover security requirements, prompt leakage risks, AI-generated vulnerability introduction, security review evidence.

5. Next Steps

We plan to:

- Publish the full PEW checklist
- Train process reviewers (Review B) on the new discipline
- Attach static analysis and CI results to all pilot merges
- Hold at least one security tabletop exercise focused on handling banned prompt content

6. A Final Note

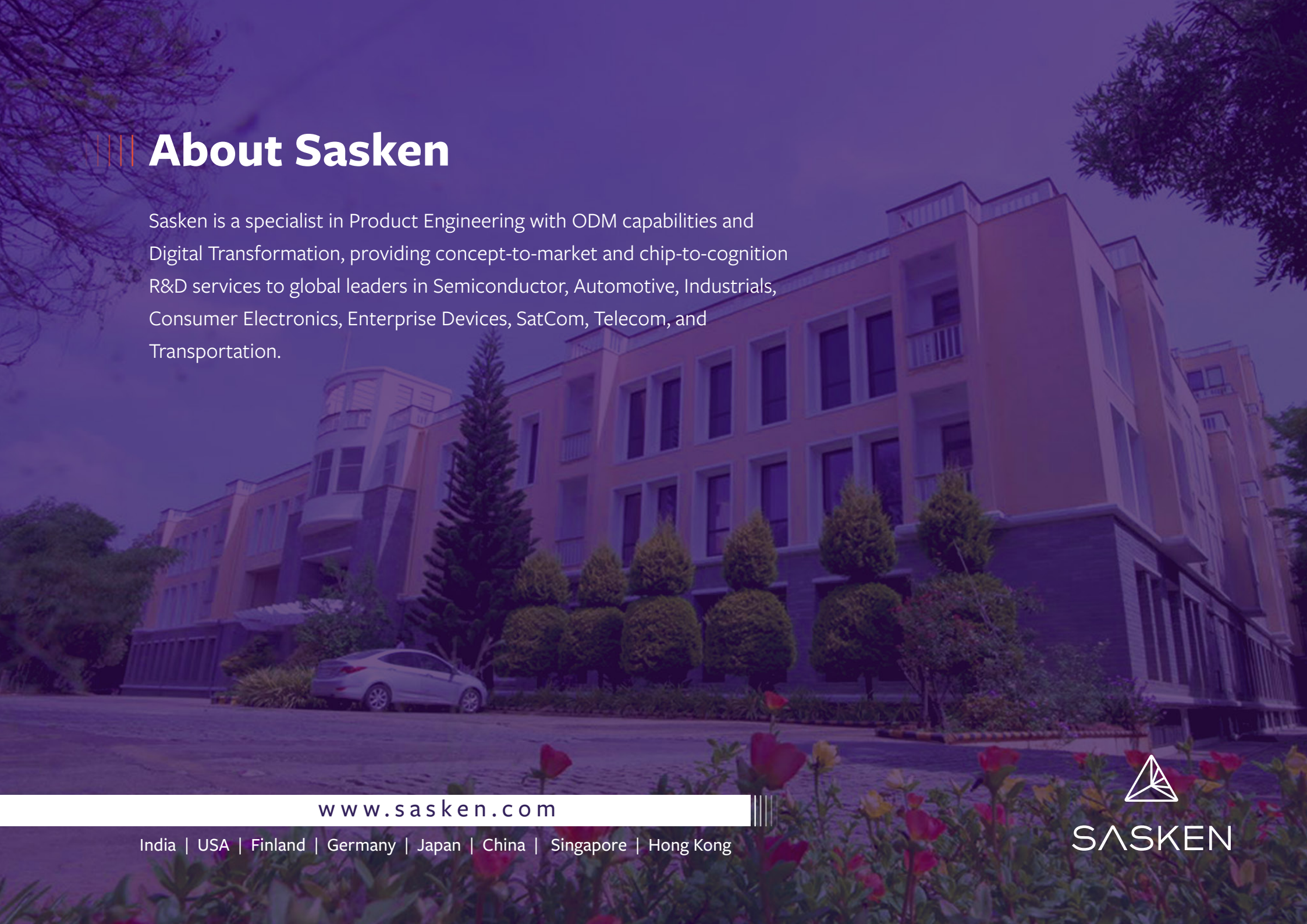
AI tools are here to stay. The question isn't whether to use them, it's how to use them responsibly in an environment that demands traceability, accountability, and auditability. PEW provides a practical path forward.

Selected references

1. S. Peng et al., “The Impact of AI on Developer Productivity: Evidence from GitHub Copilot,” Microsoft Research / arXiv:2302.06590, 2023. https://www.microsoft.com/en-us/research/publication/the-impact-of-ai-on-developer-productivity-evidence-from-github-copilot/ (accessed 2 June 2026).
2. E. Paradis et al., “How much does AI impact development speed? An enterprise-based randomized controlled trial,” arXiv:2410.12944, 2024. https://arxiv.org/abs/2410.12944 (accessed 2 June 2026).
3. Automotive SPICE Process Assessment Model (v4.0), VDA QMC
4. Stack Overflow Developer Survey (AI adoption / intent). https://survey.stackoverflow.co/2024/ai/ (accessed 2 June 2026).
5. GitHub Blog, “Does GitHub Copilot improve code quality? Here's what the data says," 2024–2025. https://github.blog/news-insights/research/does-github-copilot-improve-code-quality-heres-what-the-data-says/
6. Internal AAOS IVI reference program evidence (lint snapshots and validation notes). Available to reviewers under Sasken document control; not for external distribution without redaction.

About The Author

Debashis Panigrahi, is Senior Solution Architect at Sasken and manages the company's Automotive Accounts. He works closely with business teams on offerings, solutions, GTM strategy, and competition analysis. He has authored articles on Multimedia, Smart Devices, and Automotive Electronics in several leading trade publications.



About Sasken

Sasken is a specialist in Product Engineering with ODM capabilities and Digital Transformation, providing concept-to-market and chip-to-cognition R&D services to global leaders in Semiconductor, Automotive, Industrials, Consumer Electronics, Enterprise Devices, SatCom, Telecom, and Transportation.

www.sasken.com

India | USA | Finland | Germany | Japan | China | Singapore | Hong Kong

SASKEN