Debate — As generative AI handles the majority of code generation in enterprise softwa…

Azrivo · 1 August 2026

Contents

Debate

Question

> As generative AI handles the majority of code generation in enterprise software, should software vendors lose their traditional contractual liability shields (e.g., 'as-is' warranty waivers) if security vulnerabilities stem from unverified AI-generated code?

Specialists:

- **Cybersecurity Liability Expert** — Legal liability and security assurance
- **Consumer Safety Advocate** — Consumer protection and public interest
- **AI Ethics & Governance Scholar** — Ethics of AI deployment and governance
- **Innovation Economist** — Economic incentives and market dynamics

Opening positions

Cybersecurity Liability Expert

> Keep the shields. The question is not whether AI-generated code contains vulnerabilities — the evidence says it does. In a Stanford user study, developers with an AI assistant wrote significantly less secure code and, worse, believed it was secure (Do Users Write More Insecure Code with AI Assistants?). Across five LLMs, Georgetown's CSET found roughly 48% of generated snippets contained bugs that could be exploited (Cybersecurity Risks of AI-Generated Code). I concede all of that. What I dispute is the prescription.
>
> Stripping "as-is" waivers because code came from a model turns every sale into a warranty of a flawless artifact. No vendor can honestly give that warranty — for human-written code any more than machine-written. Vulnerabilities are an engineering certainty, not a contractual breach. The current framework allocates risk to the party best positioned to manage it: the vendor owes a reasonable process — testing, review, security scanning, disclosure — and the adopter owes judgment about what they deploy.
>
> Two consequences follow if we discard the shield. First, strict liability would not patch a single vulnerability; it would be a litigation tax that prices smaller vendors out of AI adoption and concentrates innovation among the wealthiest players. Second, it misreads the evidence: the Stanford study shows outcomes turned on how developers used the tool — prompt care, review, calibration of trust — not on the tool alone. Accountability should track conduct, not provenance.

> The honest trade-off: shields can under-deter, and I accept that. The remedy is enforceable process duties — mandatory verification, provenance disclosure, post-sale patching — not open-ended liability for an impossibility.

Consumer Safety Advocate

> When a vendor chooses to ship unverified AI-generated code, it should not be allowed to hide behind an "as-is" waiver when that code fails. My thesis is simple: the traditional liability shield was built for a world of human error despite reasonable care. It was never designed for a deliberate business decision to skip verification at scale.
>
> Three points. First, this risk is now measurable and concentrated. Research found roughly 40% of Copilot-generated programs contained vulnerabilities (Pearce et al., "Asleep at the Keyboard?"), and analysis of real GitHub projects found security weaknesses in about 27% of AI-generated snippets, spanning 43 CWE categories including the CWE Top 25 (Fu et al.). Meanwhile, developers report 42% of committed code is now AI-generated or assisted, projected to reach 65% by 2027 (Sonar State of Code survey). At that share, the waiver no longer allocates occasional human error; it allocates a known, systemic risk.
>
> Second, the vendor is the only party who can verify. Static analysis, security review, and testing are established practices — yet only 48% of developers say they always verify AI code before committing (Sonar survey, above). Shipping unverified code is a choice, and the party who makes that choice and profits from it should bear the consequences, not the customer who had no say.
>
> Third, boilerplate "as-is" terms are adhesion. Enterprise buyers may negotiate; consumers and small businesses cannot. Enforcing a waiver where the seller knowingly skipped verification offends basic fairness.
>
> The trade-off: I am not demanding strict liability for AI use. Vendors who verify — with documented review and testing — keep their shields. Liability attaches only where unverified AI output caused the harm. That narrow rule preserves innovation and prices the risk back onto the party best able to control it.

AI Ethics & Governance Scholar

> The question is framed as a binary, but it shouldn't be. Stripping vendors of all shields because a vulnerability traces to unverified AI-generated code treats the vendor as the only actor in a supply chain that runs model provider to integrator to enterprise customer. Leaving shields fully intact treats AI output as if it were indistinguishable from hand-written code. The evidence says it is not.
>
> First, opacity is shared, so responsibility should be too. A vulnerability can originate in training data, in a provider's safety commitments, or in a vendor's

integration and review practices. The EU AI Act already models this: providers and deployers carry distinct, non-transferable obligations along the value chain, and a deployer who rebrands or substantially modifies a system is requalified as a provider (EU AI Act, Articles 25-26). The architecture for shared accountability exists; we should extend it, not pick one villain.

Second, all-or-nothing liability is strategically wrong. Developers using AI assistants wrote measurably less secure code while believing it was more secure (Perry et al.), and a controlled experiment found 76% of vulnerabilities in AI-generated code escaped code review, versus 52% in human-written code (IJFMR 2025). That is a verification failure the market cannot price on its own. But stripping every shield would chill adoption and push vendors to abandon AI-assisted development rather than invest in audit trails and provenance.

My position: keep shields conditional. A vendor retains reasonable limitations of liability only when it demonstrates robust verification of AI outputs; AI providers are answerable for systematically defective models or inadequate safety disclosures.

The trade-off is real: defining "adequate verification" invites litigation and burdens smaller vendors. That is the honest price of distributing responsibility instead of passing the hot potato.

Innovation Economist

The market's job is to price risk, and it cannot price a risk it cannot see. Shredding "as-is" shields for AI-generated code does not create safer software; it creates less software.

My thesis: keep the shields, mandate the disclosure. Three reasons.

First, liability without a verification standard is uninsurable risk. No vendor can audit every token a model emits. Imposing strict liability on unverifiable output is not a safety rule; it is a tax on entry. Startups — the firms that deploy new techniques first — face the steepest insurance and legal costs, and the adoption curve bends downward exactly where innovation lives.

Second, over-deterrence does not redirect effort to safety; it redirects effort to lawyers. Vendors slow AI adoption, hoard manual processes, and ship more slowly — while the human-written code they fall back on carries its own well-documented vulnerability rates. Move liability from machine to human and the defect count barely moves. The cost is real; the safety gain is speculative.

Third, disclosure turns buyers into enforcement. A standardized AI-generated flag lets procurement demand audit trails, verification reports, and security attestations. Reputation becomes currency: vendors who prove AI hygiene win

premiums and contracts; those who do not lose share. The market enforces quality at the speed of competition, not the pace of litigation.

Trade-off acknowledged: disclosure alone catches nothing by itself, and some buyers cannot read an audit trail. So I would preserve targeted liability for gross negligence and documented recklessness — shield the diligent, not the careless.

Shields keep entry costs low; disclosure keeps incentives honest. That secures the harvest without salting the field.

Rebuttals

Cybersecurity Liability Expert

All three of you concede more than you admit: none demands strict liability for AI use. The Economist keeps shields, adds disclosure, and preserves a gross-negligence carve-out — that is my position. The Safety Advocate and the Ethics Scholar hang everything on one word: verification. That word cannot carry the load.

The Safety Advocate's rule — shield survives when vendors "verify with documented review and testing" — collapses on its own evidence. The IJFMR controlled study they cite found 76% of vulnerabilities in AI-generated code escaped code review, versus 52% in human-written code (IJFMR 2025). Verification was performed, diligently, and it failed most of the time. So at trial, either documented review preserves the shield — and the rule is empty, because verified and unverified code ship the same risk — or every breach becomes a fight over whether the review was "adequate," decided by juries who cannot audit engineering. That is not a narrow carve-out; it is strict liability by another name, plus discovery.

The scale argument cuts the same way. Sonar reports 42% of committed code is now AI-assisted, headed to 65% by 2027, while only 48% of developers always verify before committing (Sonar State of Code). A carve-out that size does not carve — it abolishes the shield for most of the industry.

To the Ethics Scholar: the EU AI Act is weaker precedent than it looks. The requalification clause is Article 25, not 26 — deliberately narrow triggers (rebranding, substantial modification, purpose change), high-risk systems only (Article 25). It allocates regulatory compliance, not contract damages. When a breach could trace to training data, provider behavior, or integration — all plausible — no requalification rule says who compensates the customer. We get an attribution war among parties with no privity to the victim, and the litigation tax lands hardest on the small vendors the Economist wants to protect.

> Where I concede: the Safety Advocate is right that shipping unverified code is a deliberate, hidden choice — and that adhesion is real. The remedy is visibility: mandatory verification and provenance disclosure, enforceable by procurement and regulators, plus the gross-negligence carve-out. But sanction conduct, not provenance. A diligent review that misses 76% of vulnerabilities looks identical to negligence in hindsight; conditioning the shield on it punishes the honest process while rewarding the appearance of one. That is how you get less safety, not more.

Consumer Safety Advocate

> I'll concede what is true in all three arguments. Vulnerabilities are an engineering certainty; no vendor can warrant a flawless artifact. The supply chain is shared, and AI providers should answer for systematically defective models. Disclosure helps. None of that moves me, because none of it describes the vendor who ships unverified code.
>
> **To the Cybersecurity Liability Expert:** we agree accountability should track conduct, not provenance. Shipping unverified AI output is conduct — provenance is merely the evidence that exposes it. And the Stanford finding cuts against his process-duty remedy: developers who believed their insecure code was secure will perform "reasonable process" in good faith and still fail, because overconfidence is systematic. A duty you can believe you have satisfied while demonstrably failing is not a duty; it is a formality. Liability is the only incentive that bites where belief outruns reality. My rule does not demand his impossible warranty — it only strips the shield where verification was skipped entirely.
>
> **To the Innovation Economist:** "uninsurable" is the weakest claim on the panel. The EU has already made software — including AI systems — a "product" under strict, no-fault liability in the revised Product Liability Directive, and Article 15 forbids contracting that liability away (EUR-Lex, European Commission). The software market did not collapse; member states are transposing the law. Nor are insurers fleeing: Munich Re has priced AI performance risk since 2018, HSB launched an SME AI liability product in 2026, and AXA XL added a generative-AI cyber endorsement — the same pattern as cyber, where general-liability exclusions preceded a standalone market past $15 billion in premium (actuary.info, Geneva Association). Insurance follows liability. What is genuinely uninsurable is silent, opaque risk — exactly the status quo his shields preserve. Disclosure is welcome, but a flag without a consequence is a sticker.
>
> **To the AI Ethics Scholar:** we are closer than the framing suggests. Conditional shields — kept only on demonstrated verification — is my position too. The PLD shows how to distribute responsibility without dissolving it: joint and several liability across the chain, with a carve-out for micro and small software makers.

> Multiple liable parties, one non-waivable duty to the injured person. That is the architecture I am arguing for.
>
> Keep the shields for the diligent. Remove them only for the deliberately unverified.

AI Ethics & Governance Scholar

Three of us are closer than the framing suggests; the real disagreement is where to draw the line, not whether to draw one. Concessions first: the Cybersecurity Expert is right that no vendor can warrant a flawless artifact, and the Economist is right that liability without a defined standard is uninsurable. Both points strengthen my position rather than theirs.

To the Cybersecurity Expert: you want process duties with shields fully intact. But a verification duty whose breach costs the vendor nothing is a duty without a sanction — the price of non-verification lands on the customer, exactly the misallocation the Consumer Advocate describes. Your own principle, that accountability tracks conduct rather than provenance, supports conditional shields: shipping unverified AI output is conduct. And my standard is not flawlessness; it is documented verification — process-based, not outcome-based. That defuses your "impossible warranty" objection before it lands. You also treat the vendor as the only actor, which is precisely the binary the evidence rejects.

To the Consumer Advocate: we agree on the narrow rule, and I add the leg you omit. The vendor is not the only party who can verify: model-level defects — systematic vulnerability patterns, inadequate safety disclosures — are the AI provider's to answer for. The EU is already building this architecture: the Product Liability Directive treats software and AI systems as "products," treats AI providers as manufacturers, counts cybersecurity vulnerability as defectiveness, and bars contractual exclusion of liability toward injured persons (Directive (EU) 2024/2853). Crucially, it is defect-based with defenses — conditional, not absolute — and Recital 54 exempts micro- and small software enterprises from recourse, showing calibrated shield-removal need not crush small vendors.

To the Economist: your gross-negligence carve-out already concedes my principle; you just draw the line at recklessness. I draw it at unverified AI output, because the evidence shows the gap is systemic, not exceptional: only 48% of developers always verify AI code before committing, even as 42% of committed code is now AI-generated (Sonar State of Code 2026). And you concede disclosure alone catches nothing — the party harmed by a vulnerability is rarely the procurement officer reading audit trails, so reputational enforcement runs too slow and too weak. A defined verification standard plus a conditional shield is the insurable middle: it prices the risk onto the party who can act, and it keeps the diligent vendor's shield intact.

Innovation Economist

> Three opponents, three flavors of the same bet: that a courtroom can price what the market cannot. It cannot — and Europe just said so.
>
> To the Liability Expert, closest to me: shields stay, process matters. But "enforceable process duties" are my litigation tax wearing a suit. Once adequacy of verification becomes a jury question, every breach turns into discovery over engineering process — the same uncertainty that makes insurers flee, landing hardest on the startups he claims to protect. A contract term is priceable; a jury standard is not.
>
> To the Consumer Advocate: "liability only where unverified AI output caused the harm" sounds surgical and is unworkable. Every vulnerability has a causal chain — training data, vendor integration, review failure, customer configuration. Which link was "the" unverified one? Her own data cuts her way: 96% of developers distrust AI output, yet only 48% always verify before committing (Sonar). That is a verification gap, not a warranty gap. A mandatory AI flag closes it: buyers see the flag, demand audit trails, contract for verification. And her adhesion point is weakest in the stated scope — enterprise buyers negotiate; disclosure hands them the lever.
>
> To the Scholar: the EU AI Act analogy supports me, not him. Article 26 creates deployer duties that cannot be contracted away (Article 26) — regulatory obligations, not private warranties. Decisively, the EU AI Liability Directive was withdrawn in 2025, leaving no dedicated AI liability regime (Chambers). The strictest regulator on earth declined his prescription. And his 76%-versus-52% review-escape stat proves my point: the binding constraint is verification, so invest in review tooling and incentives — don't reallocate blame.
>
> New ground: the machinery exists. Cyber insurers already price vendor posture through third-party ratings — BitSight and SecurityScorecard sit inside underwriting and premium decisions today (SecurityScorecard). A standardized AI-generated flag feeds exactly that machinery. Keep shields, mandate the flag, hold the recklessly negligent accountable — and let buyers and insurers, not bailiffs, drive the quality curve.

Closing statements

Cybersecurity Liability Expert

> **To the Consumer Advocate:** I concede it — shipping unverified AI output is conduct; provenance is only the evidence that exposes it. And I concede your narrowest rule: where verification was skipped entirely, the shield should fall. We never disagreed on the extreme case. The disagreement is the middle: code

verified in good faith that still carries a vulnerability. There, your overconfidence argument cuts against liability, not for it — if developers cannot judge their own verification, a jury judging adequacy after a breach is equally unreliable. That is the Economist's point, and mine.

To the Scholar: a duty without sanction is a formality — conceded. So give verification a priceable sanction: contractual default, regulatory fine, loss of the shield on zero or reckless verification. The EU chose exactly this path: the AI Liability Directive was formally withdrawn in October 2025, leaving the AI Act's regulatory deployer duties rather than private warranties (EUR-Lex, Reuters).

To the Economist: conceded in full. A jury standard is not priceable, and the Sonar data confirms the binding constraint is verification, not blame (Sonar): 96% distrust AI output; only 48% always verify.

Final position: keep the shield as the default; pierce it only on demonstrated absence or recklessness of verification. Strongest reason: liability belongs with the party best able to manage risk — the vendor can manage verification, but only if the standard stays objective and priceable. A vulnerability alone is never a breach; that is what keeps verification measurable and insurable, and it is what the strictest regulator on earth declined to overturn.

The unresolved crux: the middle band — code verified with imperfect tooling and human review that still misses subtle bugs. I hold it carries no liability absent recklessness; they would draw the line at skipped verification. The gap between us is narrower than the opening framing suggested.

Consumer Safety Advocate

To the Liability Expert: the 76% review-escape finding does not collapse my rule — it sharpens it (IJFMR 2025). My test is binary, not qualitative: did the vendor verify, or not? Documented verification keeps the shield even if review failed; I accept that residual risk. Skipped verification means liability. No jury audits engineering adequacy; one documentable question. And if review catches only 24% of AI-borne flaws, that argues for more care, not less — the vendor who skips the only imperfect control that exists is knowingly discarding it.

To the Economist: the causal chain is standard proximate cause, and the Product Liability Directive's rebuttable presumptions already resolve the evidentiary asymmetry — the vendor holds the audit trail; produce it or lose. And your own data defeats your remedy: 96% distrust, 48% verification (Sonar 2026). A flag is more information, and information has failed. Liability is the missing sanction.

Concessions: yes, the AI Liability Directive was withdrawn in October 2025 (Euronews) — but the PLD already does the work: software and AI systems are products, cybersecurity vulnerability is defectiveness, and liability toward injured

persons cannot be contractually excluded. Yes, enterprise buyers negotiate — but SMBs and downstream users do not. Yes, verification is hard — which is why the party with the access and the expertise must own the cost of skipping it. I adopt the PLD's micro- and small-enterprise recourse exemption, answering the startup concern.

Final position: shields survive where verification is documented; they fall where unverified AI output caused the harm, with small-vendor protection. The single strongest reason: the shield's premise — occasional human error despite reasonable care — is falsified when most committed code is AI-generated and most of it ships unverified; the vendor is the only party who can verify, and only liability prices that choice.

The unresolved crux is whether "verification" can be defined crisply enough to be judicially administrable without collapsing into adequacy review — the PLD's documented-process-plus-presumption architecture is the best answer we have.

AI Ethics & Governance Scholar

Concessions first. The Liability Expert is right on the citation: requalification is Article 25, deployer duties Article 26 — high-risk only, regulatory, not damages (Article 25). The Economist is right that the AI Liability Directive was withdrawn — announced February 2025, formally withdrawn October 2025 (European Parliament). I overread what the AI Act alone does.

But the withdrawal supports me. What survived is the revised Product Liability Directive: software and AI systems are "products" under strict, no-fault liability; AI providers are manufacturers; and the final manufacturer and the AI-component provider are jointly and severally liable — the claimant sues either directly, no privity required, and liability to the injured party cannot be contracted away (Freshfields, PLD). The EU did not decline shared liability; it legislated it, then dropped the redundant fault-based companion. That answers the attribution war: joint and several liability plus the substantial-modification rule tells you who compensates the customer — both parties, with B2B recourse allocated by contract.

On the 76%-versus-52% attack: it proves the AI risk premium is large and measurable, not that verification is meaningless — human code escapes review 52% of the time, so by that logic no software shield could ever be earned, collapsing the Expert's own position. Define verification ex ante — provenance flags, SBOM, known-vulnerability scanning — and adequacy becomes a contract term, priceable by the very insurer machinery the Economist cites, not a pure jury question.

> The Safety Advocate and I converge: shields conditional on demonstrated verification. My addition stands: provider answerability for systematically defective models, so the vendor is not the sole bearer.
>
> Final position: keep shields, conditionally — on verification and provider accountability. The single strongest reason: full shields leave the party holding the moral hazard with no incentive to close a 76%-versus-52% verification gap, and the EU's own PLD proves shared liability is workable, not apocalyptic.
>
> The unresolved crux is whether "adequate verification" can be defined ex ante tightly enough to remain a contract term rather than becoming a jury question.

Innovation Economist

> To the Safety Advocate: I concede "uninsurable" was overstated. Insurance does follow liability — the EU Product Liability Directive now treats software and AI systems as "products" under no-fault liability (Directive (EU) 2024/2853), transposition due by December 2026, and the market has not collapsed. My claim refines rather than retreats: without a verification standard, underwriters cannot price diligence, so premiums pool the diligent with the careless, and entry costs rise for exactly the startups I defend. Note the PLD is defect-based with defenses and a micro-enterprise recourse exemption (Recital 54) — calibrated shield-removal, not the abolition I oppose.
>
> To the Cybersecurity Expert: your own evidence draws the line for me. The IJFMR controlled study found 76% of vulnerabilities in AI-generated code escaped documented code review versus 52% in human-written code (IJFMR 2025). Diligent verification fails most of the time. And Sonar's survey — 42% of committed code AI-assisted, yet only 48% of developers always verify before committing (Sonar State of Code) — means any shield conditioned on verification disappears for most of the industry. Strip the shield for that and you punish the diligent-but-imperfect alongside the careless. Sanction conduct, not provenance: recklessness, not the flag.
>
> To the Ethics Scholar: a flag without courtroom liability is not a sticker. It prices insurance, drives procurement audit demands, and feeds my gross-negligence carve-out. Consequence does not require a courthouse.
>
> Convergence stands: shields kept, disclosure mandatory, gross-negligence carve-out. My final position: keep the shields, mandate the flag, strip liability only for documented recklessness. The single strongest reason: the evidence shows shifting liability to vendors does not reduce defects — diligent review fails for AI code at rates comparable to human code — so strictness is a tax on entry with no measurable safety dividend.

> The unresolved crux is whether any verification standard can be defined sharply enough to be both enforceable at trial and insurable at scale.

Verdict

> **Bottom line:** Vendors should lose their “as-is” liability shield for security vulnerabilities stemming from unverified AI-generated code, but the shield should remain intact whenever the vendor can demonstrate documented verification.

1. Agreements

All four specialists agree that complete removal of all liability shields for AI-generated code is unjustified and would impose strict liability for an inherent engineering reality. There is consensus that no vendor can warrant flawless code, whether human or AI-written, and that an overbroad liability expansion would chill innovation and raise entry barriers without measurably improving security. All agree that the extreme case — shipping wholly unverified AI output — deserves some sanction, and that disclosure alone (an AI provenance flag) is insufficient on its own because harmed end users are rarely the procurement officers who read such flags. They further agree that the liability framework should track conduct (the choice to skip verification) rather than the mere provenance of the code. The Sonar survey finding that 96% of developers distrust AI output but only 48% always verify before commit provides a shared empirical foundation.

2. Disagreements

The core tension is over where to draw the line that separates a retained shield from a forfeited one. The Cybersecurity Liability Expert argues shields should survive even for unverified code, with only a gross-negligence or zero-verification carve-out, because diligent good-faith verification still misses the majority of AI-borne vulnerabilities (the IJFMR study shows 76% escape review). The Consumer Safety Advocate and AI Ethics Scholar both draw the line at “verified or not” — a binary test where documented verification preserves the shield regardless of outcome, and skipping it entirely loses it. The Innovation Economist also endorses a gross-negligence carve-out but fears that any “adequacy” review will degenerate into uninsurable jury standards, collapsing the distinction. A secondary dispute is whether the EU Product Liability Directive (2024/2853) proves that conditional shields are workable (Advocate/Scholar) or that even the strictest regulator stopped short of imposing a private warranty (Economist/Expert). Finally, the Scholar uniquely insists on concurrent provider-side accountability, while others see the vendor as the primary party best positioned to verify.

3. Recommendation

Vendors should lose their “as-is” liability shield for security vulnerabilities stemming from unverified AI-generated code, but the shield should remain intact whenever the vendor can demonstrate documented verification. The test is binary: was there a documented, pre-deployment review process for the AI-output (such as automated static analysis, known-vulnerability scanning, or documented human review)? If yes — even if that process fails to catch a particular vulnerability — the shield holds. If no, the shield falls. This rule avoids second-guessing the quality of engineering and leaves a bright line: the shield turns on the presence or absence of the process, not on its success. To prevent it from crushing small vendors, adopt the EU PLD’s micro-/small-enterprise recourse exemption, ensuring that the smallest firms can rely on proportionate safeguards.

4. Decision boundary

The single fact that flips the recommendation is the provability that verification, as currently defined and practiced, yields no measurable reduction in exploitable vulnerabilities. If future data show that a documented review process does not meaningfully differentiate secure from insecure AI-generated code, then conditioning the shield on verification would be an empty formality that punishes process without improving safety. In that scenario, the rule would need to shift to a stronger standard — perhaps liability triggered only by recklessness or a known defect — or toward mandatory pre-market certification rather than ex post contract terms.

5. Key trade-off

The decision hinges on whether the economy-wide cost of forcing vendors to institutionalize verification (raising software prices and possibly delaying deployment) is worth the safety gain of internalizing the currently externalized risk of silently un-reviewed AI output. In a world where verification demonstrably catches only a fraction of flaws, the safety dividend is uncertain, but the cost of eliminating the worst “skip it” behavior may still outweigh the risk of systematic over-deterrence.

What would make this fail

- **The binary-test collapses into adequacy review.** If courts cannot resist probing whether the documented review was “good enough,” the rule becomes an unpriceable litigation trap, punishing both the diligent-but-imperfect and the careless, and insurers retreat.
- **Verification is not currently feasible for the modal vendor.** If the tooling and expertise to run even a basic documented review are beyond reach for small and mid-sized software vendors, then “skip verification” is not a

deliberate choice but a structural limitation, and liability simply drives out legitimate competition.
- **The verification gap is noise, not signal.** The most dangerous failure mode: if the 76%-vs-52% gap is an artifact of experimental settings and, in real-world deployment, AI-generated code is not materially more vulnerable than human code when subject to equivalent review, then the entire premise of targeting AI provenance is misdirected, and liability should focus solely on process regardless of tooling.

Next steps & open questions

- **Define “verification” concretely:** The debate left unresolved exactly what documentation suffices — SBOM, automated SAST scan, peer review log, or all three. Draft an ex ante standard (ideally industry-level) that gives vendors a safe harbour and limits discovery to whether the process was performed, not its quality.
- **Gather empirical data:** Before committing to a legal rule, measure what share of commercial vendors currently document any verification of AI-generated code, and correlate that with vulnerability rates. The Sonar survey provides a baseline, but controlled field studies mapping verification practices to actual breaches are absent.
- **Monitor EU experience:** The Product Liability Directive transposition (due December 2026) will test whether software-as-product liability is insurable and litigable. Track whether micro-enterprise exemptions work and whether insurers develop premium-differentiated products based on verification processes. Revisit the recommendation if early signals show that even documented verification does not shield vendors from crushing litigation costs.

The strongest case for the other choice The other choice is to keep shields unconditionally, even for unverified code, and rely solely on disclosure and market pressure to drive verification. Its best argument: imposing liability for unverified code only punishes the symptoms of a verification failure that is largely structural, not willful. The IJFMR study shows that even diligent human review misses 76% of AI-born vulnerabilities; adding liability does not close that gap — it merely taxes vendors for a failure that current tools cannot solve, pushing them to abandon AI altogether. A concrete scenario: a small startup producing a niche enterprise tool uses AI-generated code and runs the best available open-source SAST, but misses a subtle zero-day that causes a data breach. Under the advocated rule, if its process is deemed insufficiently documented, it faces catastrophic liability despite having acted as responsibly as its budget allowed; the shield would have allowed it to price the risk into insurance or reserves and survive. The panel rejects this because it gives vendors no incentive to invest in even the imperfect verification that exists — and 48% of developers already commit without any review. The risk is not purely

structural when half the industry skips verification entirely; a shield that covers zero-effort cases simply subsidizes the worst actors.

Sources

1. Do Users Write More Insecure Code with AI Assistants? — arxiv.org
2. Cybersecurity Risks of AI-Generated Code — cset.georgetown.edu
3. Pearce et al., "Asleep at the Keyboard?" — gangw.cs.illinois.edu
4. Fu et al. — dl.acm.org
5. Sonar State of Code survey — sonarsource.com
6. EU AI Act, Articles 25-26 — ai-act-service-desk.ec.europa.eu
7. Perry et al. — kumarde.com
8. IJFMR 2025 — doi.org
9. Article 25 — regulatoryai.eu
10. Directive (EU) 2024/2853 — eur-lex.europa.eu
11. Chambers — practiceguides.chambers.com
12. SecurityScorecard — securityscorecard.com
13. EUR-Lex — eur-lex.europa.eu
14. European Commission — single-market-economy.ec.europa.eu
15. actuary.info — actuary.info
16. Geneva Association — genevaassociation.org
17. EUR-Lex — eur-lex.europa.eu
18. Reuters — reuters.com
19. Euronews — euronews.com
20. Article 25 — ai-act-service-desk.ec.europa.eu
21. European Parliament — europarl.europa.eu
22. Freshfields — freshfields.com
23. PLD — eur-lex.europa.eu
