

Accepting Software Delivered by Provider Coding Agents

A Buyer Acceptance Method for the Released Product

Phil Hatch | Akholi

October 23, 2026

Software sourcing managers should accept the released software and its evidence, regardless of how much code agents generated. The supplier must demonstrate secure development, review, testing, provenance, deployment, and support for the actual delivered version.

Accept the Product Rather Than the Generated Code Count

The buyer delivery manager should define the software behavior, quality, security, and operating conditions required for acceptance. Agent-generated code can accelerate production while leaving design, review, integration, and maintenance work unresolved. The percentage written by agents is not evidence that the product meets its obligations.

NIST's SSDF provides a common secure-development vocabulary for producers and purchasers.¹ It is not a certification of a coding agent. The buyer should use appropriate development disciplines to assess the released product and the supplier's process, including the work performed by people to make generated code suitable for use.

Start with the actual release candidate. Identify source, dependencies, build configuration, test results, deployment instructions, and support commitments. A demonstration from a different branch or environment should not close acceptance of the version the buyer will operate.

¹ Murugiah Souppaya et al., "Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities," National Institute of Standards and Technology, February 2022, SP 800-218, abstract and acquisition use, <https://csrc.nist.gov/pubs/sp/800/218/final>.

Preserve the Human and Agent Contribution

The study's manual-alteration reports show why human repair can sit behind an automation claim.² They do not measure coding-agent defect rates. In software delivery, the buyer should require visibility of material review, correction, and unresolved limitations without treating every human contribution as a failure.

Record who approved architecture, reviewed consequential changes, resolved security findings, and authorized release. Preserve the relevant artifact and evidence versions. An agent-written test that passes an agent-written implementation is useful information, but it should not be the sole independent evidence that the business requirement was met.

For a hypothetical account-access feature, tests should show both the permitted path and denial of unauthorized operations. A correct-looking interface cannot establish authorization enforcement. The acceptance plan should follow the consequences of the software, including failure, recovery, and misuse conditions.

Use a Release Evidence Schedule

Figure 1 proposes four evidence areas: functional behavior, secure production, provenance, and operability. Link each to the delivered version and an accountable supplier owner. The buyer should be able to identify unresolved findings and the authority that accepted them, rather than receive a general assurance that testing was completed.

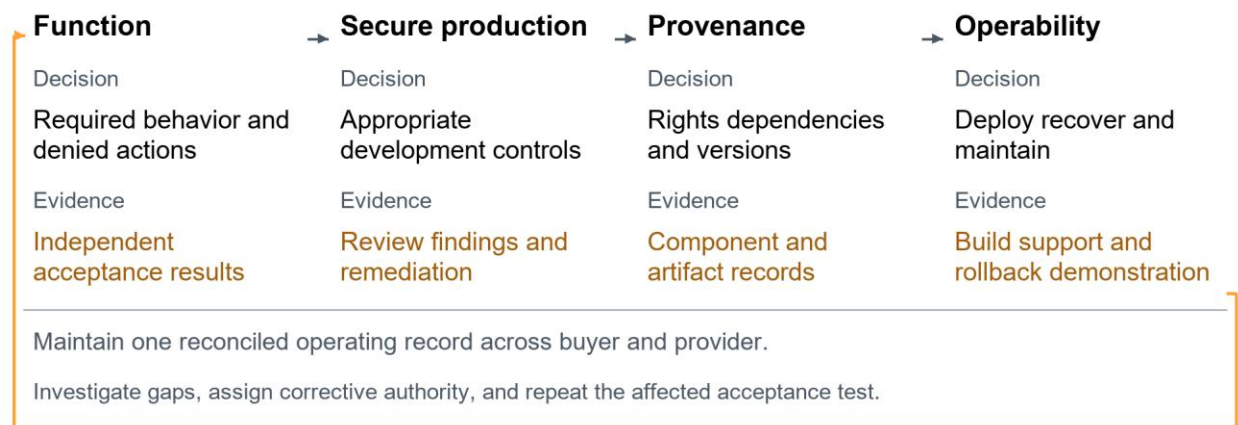
The UK Software Security Code is voluntary guidance addressing supplier development and maintenance practices.³ The buyer should select requirements appropriate to the product and record how the supplier demonstrates them. A self-assessment should be evaluated within its scope, not treated as proof that vulnerabilities are absent.

Test the deployed configuration with representative data and integrations. Include dependency failure, access restrictions, rollback, and the handoff to support. Code that runs in the supplier's environment may still require undocumented services, credentials, or licenses that the buyer cannot lawfully or practically reproduce.

² Phil Hatch, *Agentic AI Outsourcing at Tier 1 Financial Institutions: Full Study Report*, version 1.8.2 (Akholi, forthcoming manuscript dated November 1, 2026), Part 5, "The Five Manual Alteration Methods" and "Disclosure of Manual Alteration," table 5.8, <https://doi.org/10.2139/ssrn.7164559>.

³ UK Department for Science Innovation and Technology, "Software Security Code of Practice," UK Department for Science Innovation and Technology, updated January 15, 2026, voluntary scope and software security and resilience principles, <https://www.gov.uk/government/publications/software-security-code-of-practice/software-security-code-of-practice>.

Figure 1. The coding agent release evidence schedule



Author's proposed method; adapt to the service and applicable jurisdiction.

Source: Phil Hatch, proposed framework developed for this paper.

Resolve Rights and Continuing Support

Counsel and technical leads should assess licenses, third-party components, generated outputs, confidential inputs, and permitted use of the buyer's code for further training or reuse. A supplier warranty should describe the actual rights and remedies; it should not be replaced by a broad assertion that AI-created material is free of restrictions.

EU, UK, Swiss, and US products may face different security, privacy, accessibility, sector, and software obligations. The acceptance plan should identify the requirements applicable to the intended use and market. Agent generation does not remove the need for that assessment.

Define the supplier's duties for defects, vulnerabilities, dependencies, and updates after acceptance. Preserve enough documentation and build evidence for another qualified team to maintain the product where the contract requires it. The buyer should not accept a deliverable that only the original agent configuration can understand or rebuild.

Release Only the Version Supported by Evidence

A reasonable objection is that a full review of generated code can consume the productivity gain. Use risk-based review and automated checks, while preserving independent tests of material behavior and security boundaries. The correct depth depends on consequence; it should not be determined solely by the speed at which code can be produced.

The manager should reconcile the acceptance criteria, evidence, unresolved findings, rights, and operating handoff before release. If the supplier changes the code after testing, determine which evidence must be repeated. A prior clean test cannot automatically approve a different artifact.

Accept the software when the delivered version performs the required function, its material risks are understood, and it can be operated and supported under the agreed conditions. Coding agents change how the supplier produces the work. They do not change the buyer's need for a defensible product acceptance decision.

Evidence and Source Note

The study is a nonrepresentative cross-section of 850 vendor delivery-team respondents on 150 engagements serving Tier 1 financial institutions, collected during January–June 2026. Neither banks nor vendors corroborated the accounts. It establishes reported experience, not causation, legal breach, or market prevalence. Its application here is a mechanism to investigate, not an estimate of conditions in other industries. The author-supplied version 1.8.2 is dated November 1, 2026 and is cited as a forthcoming manuscript consulted October 23, 2026.⁴

This is an analytical decision paper based on the supplied study and selected primary public sources checked through October 23, 2026. The original contribution is a released-product acceptance schedule that connects coding-agent output to independent functional tests, secure development, provenance, and continuing operability. The proposed method has not been independently validated. Legal requirements depend on the entity, activity, jurisdiction, and effective provisions; supervisory observations, research, and professional commentary have distinct authority.

Bibliography

Hatch, Phil. *Agentic AI Outsourcing at Tier 1 Financial Institutions: Full Study Report*. Version 1.8.2. Akholi. Forthcoming manuscript dated November 1, 2026. Author-supplied manuscript consulted October 23, 2026. <https://doi.org/10.2139/ssrn.7164559>.

Souppaya, Murugiah, Karen Scarfone, and Donna Dodson. "Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities." National Institute of Standards and Technology. February 2022. <https://csrc.nist.gov/pubs/sp/800/218/final>.

UK Department for Science Innovation and Technology. "Software Security Code of Practice." UK Department for Science Innovation and Technology. Updated January 15, 2026. <https://www.gov.uk/government/publications/software-security-code-of-practice/software-security-code-of-practice>.

⁴ Hatch, *Agentic AI Outsourcing*, Introduction, "Executive Summary" and "Evidentiary Basis and Its Limits."