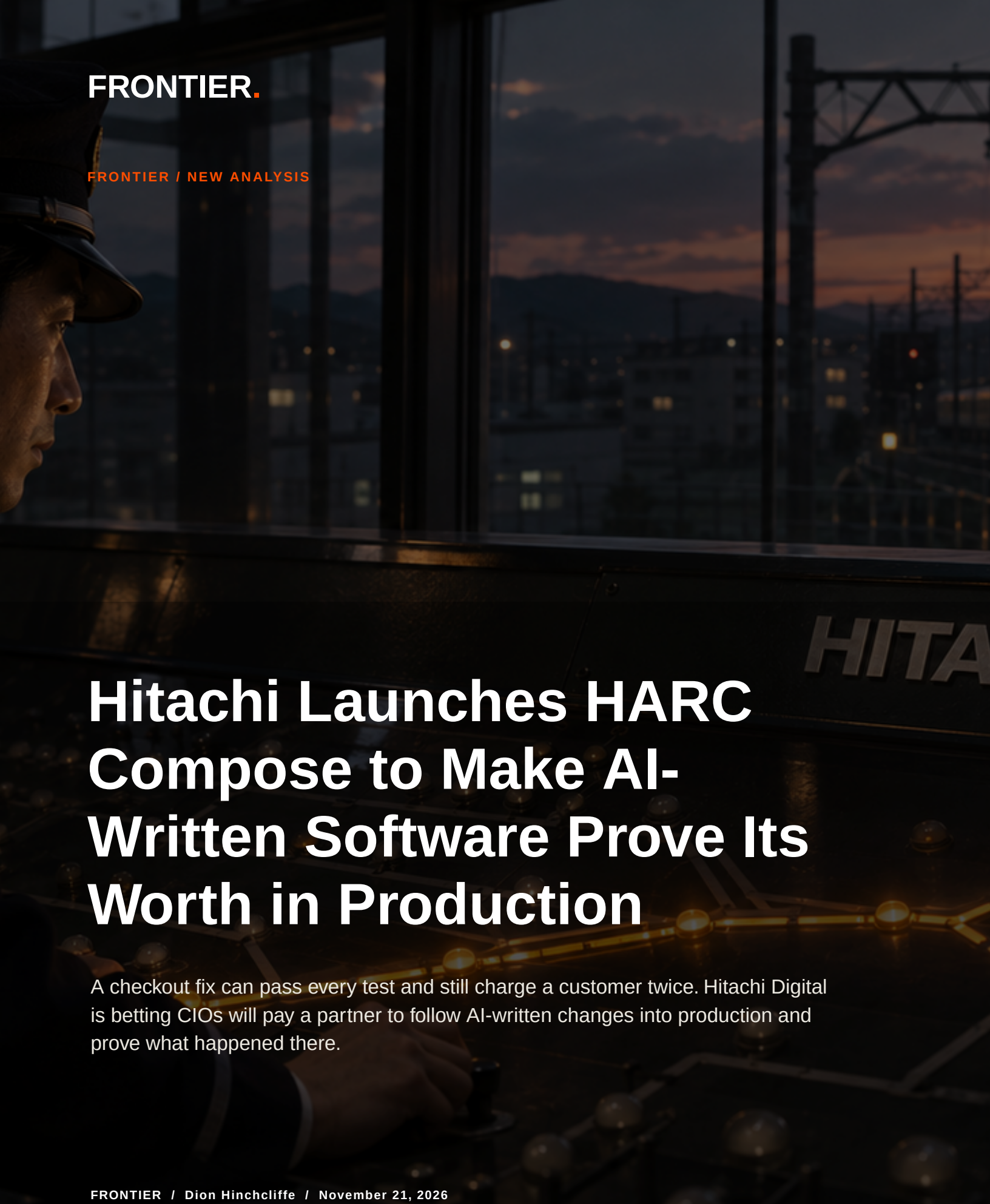
FRONTIER.

FRONTIER / NEW ANALYSIS



Hitachi Launches HARC Compose to Make AI-Written Software Prove Its Worth in Production

A checkout fix can pass every test and still charge a customer twice. Hitachi Digital is betting CIOs will pay a partner to follow AI-written changes into production and prove what happened there.

FRONTIER / Dion Hinchcliffe / November 21, 2026

NEW ANALYSIS

Built to change a decision.

A checkout fix can pass every test and still charge a customer twice. Hitachi Digital is betting CIOs will pay a partner to follow AI-written changes into production and prove what happened there.

EDITION	**New Analysis**
AUTHOR	**Dion Hinchcliffe**
PUBLISHED	**November 21, 2026**

This PDF is a publication rendering of the canonical Frontier News article. The web article remains the live source of truth; analysis, figures, and source citations travel with this edition.

What changed

Hitachi Digital launched HARC Compose on November 18 with an unusual definition of when an engineering job is finished. An AI agent can write the change, a reviewer can approve it and operations can put it into production. Under Hitachi's approach, the work stays open until the team can demonstrate that the change delivered the business result it set out to achieve. The offering combines software with Hitachi engineers who work alongside the customer. Hitachi says it is available worldwide, with deployments underway in finance, healthcare and technology. Its announcement gives no customer names, prices or measured results from those deployments.[1]

Consider a retailer losing sales whenever its payment processor takes too long to reply. An agent could change checkout to try the payment again. More shoppers might get an order confirmation. A few might also get charged twice because the original payment succeeded even though its response arrived late. A working checkout and an accurate customer bill are two different things. Which one did the retailer pay its software team to deliver?

That hypothetical is a useful way to read Hitachi's announcement. CIOs already buying coding assistants face a decision about the next dollar. It could buy more capacity to produce software, or it could help the people who run that software establish what it did to the business. HARC Compose is a bid for the second budget.

Why it matters

DORA's research explains why this distinction matters now. Researchers found that greater AI adoption was associated with both higher software delivery throughput and greater instability. In a separate analysis of 1,110 written responses from Google engineers, time saved generating code frequently resurfaced as time spent auditing and verifying it. Those findings describe a tension in software delivery. A CIO evaluating Compose still needs evidence that it can improve those economics on the enterprise's own systems.[2]

Hitachi's design starts with the information an agent needs before it changes anything. Its Context-Aware Knowledge Engine, or CAKE, draws on company policies, system knowledge, previous decisions and production history. Hitachi says it flags gaps and contradictions. An orchestration engine coordinates the customer's chosen agents across development, deployment and operations. Teams can adopt individual capabilities and keep their existing tools.[1]

In the checkout example, the most valuable piece of enterprise knowledge might be a note from the payments team explaining why it disabled retries two years ago. An agent reading today's ticket and source code alone could faithfully recreate yesterday's incident. Surfacing the note gives an engineer a specific question to settle with the processor. It also exposes a cost that belongs in the business case. Someone at the retailer still has to decide which policy is current and who has authority to change it.

Hitachi also brings an industrial engineering practice into the development process. With failure mode and effects analysis, teams identify how a change could go wrong and use those scenarios to choose tests, controls and recovery steps. Guided work calls for engineer review before an agent acts. In autonomous work, engineers monitor agents acting within agreed limits. Hitachi's CTO says people retain oversight of releases.[1] For a payments owner, those limits should distinguish the authority to propose a retry, the authority to enable it for live traffic and the authority to issue a refund. The financial consequences differ even when all three actions involve the same application.

There is a useful distinction here for buyers comparing governance products. Frontier's coverage of BlueFlag's expansion examined whether a team could tell when an agent approved a change without a human reviewer.[3] Compose raises the question that survives even a perfectly documented approval. Did the change improve anything? A release record can name the person who signed off on checkout. It takes transaction and settlement data to tell that person whether customers paid the right amount.

Frontier take

The most interesting part of HARC Compose is the services agreement a CIO might now be able to negotiate. Enterprises routinely hand a business problem to one team, the build to another and the pager to a third. Each can meet its own commitments while the original problem persists. A partner willing to stay through production gives a CIO a chance to close that gap by writing a better acceptance clause. For the retailer, payment completion and duplicate settlement would appear together in the agreement. Both the engineering director and the payments owner would know what evidence closes the job.

Hitachi has a commercial reason to make that offer. In October it brought GlobalLogic and Hitachi Digital Services together under the Hitachi Digital brand. On its earnings call, management described GlobalLogic's strength in product and data engineering, its sister company's strength in enterprise platforms and cloud operations, and the opportunity to sell more services to existing customers. The Compose announcement explicitly connects its embedded engineers to recurring service revenue. A CIO can use that alignment to secure one partner accountable across build and run. Procurement should also price how much of that accountability still depends on those engineers being present next year.[4][5][1]

The harder problem is deciding who gets to declare success. A holiday promotion, a different mix of payment cards or an improvement at the processor could all lift checkout completion in the same week the new code goes live. Google's Site Reliability Workbook describes how to compare a limited release with a control, choose representative traffic and select measures that can reasonably be attributed to the change.[6] A CIO has good reason to insist on access to the underlying evidence. If finance and engineering cannot independently reproduce the result, the supplier effectively gets to grade its own work.

The same test applies when things go wrong. Reverting a software release can stop another duplicate charge. It does nothing for the customer who has already been billed twice. Someone must identify the affected transactions, reconcile the ledger and get the money back. That is where a promise to own the outcome becomes valuable, and where a vague one becomes expensive. The question for a service review is simple enough to ask aloud. Who still owns this incident after the dashboard turns green?

Three recommendations for CIOs

1. Make the first purchase contingent on one business outcome you can independently measure. Choose a service with an identifiable owner, a baseline and enough traffic to compare results. For checkout, agree in advance on successful orders, duplicate settlements and the observation period needed to see both. Track engineering time, service fees, change failures and recovery time alongside them. Expand the pilot only when the business owner can reproduce the improvement and its cost. This gives finance a defensible answer when the first productivity claim arrives.[7][6]
2. Use a past incident to decide where agents earn autonomy. Before giving an agent permission to change a live service, replay an incident in a test environment with the original ambiguity intact. Have the team show which historical decision surfaced, who resolved conflicting instructions and which action required human approval. Then interrupt the exercise and ask the on-call engineer to recover the service. Keep that class of change in guided mode until the customer's team can do it. The exercise makes the proposed limits observable before they carry financial consequences.[1]
3. Put the handover in the contract before the second team signs up. Require customer access to decisions, approval history, test evidence and production measurements in a usable format. Name the party responsible for restoring service and the one responsible for correcting customer records. At the pilot exit, have an engineer who did not build the change reconstruct it from that material without the supplier's help. Make a successful handover a condition for extending the agreement. That lets the CIO buy Hitachi's expertise while retaining the ability to run the business when a different engineer answers the pager.

SOURCES / CITATIONS

Trace the argument.

Every source below is linked. Source numbering follows first appearance in the report.

01 **Hitachi Digital's HARC Compose launch announcement, November 18, 2026**
https://www.hitachidigital.com/about/news-room/press-release/hitachi-digital-launches-harc-compose-to-turn-enterprise-intent-into-trusted-outcomes-at-ai-speed/

02 **DORA research on AI adoption, delivery throughput and instability**
https://dora.dev/insights/balancing-ai-tensions/

03 **Frontier Research on human approval and AI agents in software delivery**
https://www.thisisfrontier.ai/news/blueflag-expands-ai-agent-governance-as-software-controls-move-beyond-human-developers

04 **Hitachi's announcement uniting GlobalLogic and Hitachi Digital Services**
https://www.hitachidigital.com/about/news-room/press-release/hitachi-launches-digital-services-company/

05 **Hitachi's October 28 earnings Q&A on the economics of the integration**
https://www.hitachi.com/content/dam/hitachi/global/en/press/files/2026/10/261028/261028_summary.pdf

06 **Google's Site Reliability Workbook on evaluating a limited rollout**
https://sre.google/workbook/canarying-releases/

07 **DORA's definitions of change failure and deployment recovery measures**
https://dora.dev/guides/dora-metrics/

FRONTIER.

TAKE THE NEXT STEP

Push this further.

Research is useful when it changes a decision. Bring us the question, the constraint, or the argument you want tested.

RESEARCH / BRIEFINGS

inquiries@thisisfrontier.ai

Briefings, advisory work, executive discussions, and research inquiries.

REPRINTS / LICENSING

reprints@thisisfrontier.ai

Reprints, enterprise distribution, excerpts, and licensing.

FRONTIER

www.thisisfrontier.ai

Research, Data Insights, News, and the live Frontier corpus.

All rights reserved. Contact Frontier for redistribution or reprint rights.

SEE FURTHER. MOVE SOONER.