

EXECUTIVE PAPER · SYSTEMS RESCUE

Measure Delivery the Way You Measure Sales

Leadership reviews the sales pipeline every month with numbers everyone understands. Software delivery deserves the same review, and the measures already exist.

AUTHOR

Chris Watkins

SERVICE

Systems Rescue

FOR

CEO, CFO, CTO · 10 min read

AT A GLANCE

The problem. Leadership judges most engineering organizations by headcount, project status colors and how busy people look. None of those tell you how fast a change reaches customers or how often it breaks something. AI coding tools have made the gap worse: developers in a 2025 trial felt 20% faster while measuring 19% slower.

Why it persists. Executives lack a shared vocabulary for delivery, so engineering reports activity because activity is easy to count. Earlier attempts to measure developers individually caused gaming and resentment, which made many engineering leaders wary of any measurement at all.

The recommendation. Review five delivery measures monthly for each critical service: deployment frequency, lead time for changes, change fail rate, recovery time and rework rate. Set targets as direction and thresholds per service, review trends the way you review pipeline, and never rank individuals on them. Judge AI tools by what they do to these numbers.

The Decision in Front of You

A sales leader who walked into the monthly review with "we made 4,000 calls and the team is working hard" would be sent back to get the pipeline. Engineering reviews often sound exactly like that. They report tickets closed, story points burned, projects at green and the number of people on the team.

The decision is whether leadership will review software delivery with outcome measures, on a fixed cadence, the way it reviews revenue. The alternative is to keep funding engineering on headcount and confidence.

The stakes are larger in 2026 than they were five years ago. In the 2025 DORA survey of nearly 5,000 technology professionals, 90% reported using AI at work and more than 80% believed it had made them more productive [4]. Most companies are now spending on AI coding tools without a baseline that would show whether delivery improved.

What's Happening

The measures exist and have a decade of research behind them

Google's DORA research program, summarized for executives in *Accelerate* in 2018, built its measures from years of surveys tying software delivery to organizational performance [2]. The current set has five measures in two groups [1]:

- **Throughput.** Lead time for changes, from commit to running in production. Deployment frequency. Recovery time after a failed deployment.

- **Instability.** Change fail rate, the share of deployments that need a rollback or hotfix. Rework rate, the share of deployments that are unplanned and triggered by a production incident.

DORA's finding across its research is that speed and stability move together: the best teams do well on all five [1]. That finding matters to a CFO. It means the usual argument that quality requires slowing down, or that speed requires accepting breakage, does not hold up in the data.

Perception and measurement have split apart

METR ran a randomized controlled trial in early 2025 with 16 experienced open-source developers on 246 real tasks. With AI tools allowed, tasks took 19% longer. The developers expected to be 24% faster beforehand and believed afterward they had been 20% faster [6]. A February 2026 follow-up produced estimates ranging from an 18% speedup to a 9% slowdown for returning developers, and METR described its own data as weak evidence because many developers would no longer work without AI [7].

The lesson for leadership is narrower than "AI does not help." Self-reported productivity is unreliable in both directions, and only measured delivery tells you what changed.

The organizational data shows the same pattern. DORA's 2024 report associated rising AI adoption with an estimated 1.5% fall in delivery throughput and a 7.2% fall in stability [3]. In 2025, DORA found AI adoption associated with higher throughput for the first time, while its association with instability continued [4]. DORA's summary is that AI amplifies an organization's existing strengths and weaknesses [5]. GitClear's analysis of 211 million changed lines found copy-pasted code rising from 8.3% to 12.3% of changes between 2021 and 2024, while refactoring fell from 25% to under 10% [10]. A team can ship more and break more at the same time, and only the instability measures will show it.

Two recent cases

Sonos, 2024. In May 2024 Sonos released a rebuilt mobile app that shipped with missing features and defects its customers noticed at once; on its next earnings call the company put the short-term cost of fixing the app and winning customers back at $20 million to $30 million [11]. The CEO stepped down in January 2025, and Sonos forecast first-quarter sales down 9% to 22% from the year before [12]. (The paper *Stop the Bleeding Before You Modernize* covers the rewrite itself.) A release judged by its launch date shipped a large batch of change with no fast path back. Change fail rate and recovery time, reviewed monthly on that product, would have put the risk in front of leadership before customers found it.

AI usage as a performance measure, 2025. In April 2025 Shopify's CEO published a memo saying AI usage would factor into performance reviews, and that teams must show AI cannot do the work before requesting more headcount [13]. Usage is an input. It tells leadership the tools are being opened. Given DORA's finding that AI adoption is associated with rising instability [4], a usage target without delivery measures beside it can reward a team for producing more change while its failure rate climbs. Usage belongs on the dashboard next to the outcomes it is supposed to improve.

CrowdStrike's July 2024 outage, covered in other papers, is the reminder of what a missing control on change costs: its root cause analysis lists the absence of staged deployment for the content update that crashed Windows hosts [14].

Why the Usual Approach Fails

The usual measures count effort. Headcount measures cost. Story points are a team's private estimate of size, and they inflate as soon as anyone outside the team watches them. Tickets closed and lines of code reward splitting work and writing more of it. None of them say whether a customer received anything.

The 2023 debate over a McKinsey proposal to measure individual developer productivity shows the other failure. Kent Beck and Gergely Orosz argued that four of the five new metrics McKinsey proposed measured effort or output, and that measuring those invites gaming and damages engineering culture [9]. The SPACE framework, published by Nicole Forsgren and colleagues, reached a similar conclusion in 2021: productivity cannot be captured by one metric, and activity levels alone are insufficient [8]. Many engineering leaders took the lesson to mean that executives should not measure engineering at all.

That leaves a vacuum. Leadership still has to decide how much to spend on engineering, and without outcome measures it falls back on headcount and gut feel. DORA's own guidance resolves the tension: apply the measures to services and teams, use several measures that pull against each other, and use them to track each team's own improvement over time [1].

The Framework: The Delivery Pipeline Review

A sales pipeline review works because everyone agrees on the stages, the numbers come from the system of record, and the conversation is about trend and exceptions. Delivery can be reviewed the same way.

Exhibit 1. Each sales review habit has a delivery equivalent

Sales pipeline review	Delivery pipeline review	What it tells leadership
Sales cycle length	Lead time for changes	How long a finished change waits before customers get it
Deals closed per month	Deployment frequency	How often value reaches customers
Deals lost after signature	Change fail rate	How often a release hurts customers
Time to rescue an at-risk account	Recovery time after a failed deployment	How long customers feel a bad release
Discounts and concessions to save deals	Rework rate	How much capacity goes to unplanned repair
Calls and meetings logged	Commits, story points, AI tool usage	Activity; useful context, never the headline

Source: author's framework; delivery measures as defined by DORA [1].

Four rules keep the review honest.

Per service. Measure each critical service on its own. DORA warns that comparing a mobile app with a mainframe system misleads [1]. A ledger that deploys monthly may be healthy; a customer website that deploys monthly usually is not.

Targets as direction and thresholds. DORA cautions against turning these measures into goals, because goals get gamed [1]. Set each service a direction ("lead time down by a quarter this year") and a threshold that triggers a conversation ("change fail rate above 15%"). Throughput targets always sit beside instability targets, so nobody can hit one by sacrificing the other.

Teams only. No individual rankings, no pay tied to these numbers. That is the line the McKinsey debate drew, and it is the line that keeps engineers reporting failures honestly [9].

From the system of record. The numbers come from deployment pipelines and incident tools automatically. A number someone types into a slide gets negotiated.

What rework costs

Exhibit 2. Illustrative cost of unplanned work for a 100-person engineering organization

Assumption or result	Today	After one year of review
Engineers	100	100
Fully loaded cost per engineer	$180,000	$180,000
Share of time on unplanned repair	20%	10%
Annual cost of unplanned repair	$3.6M	$1.8M
Capacity returned to planned work		$1.8M

Source: author's illustrative estimate. Share of time on unplanned repair is an assumed figure for a hypothetical organization; measure your own from incident and ticket data. Rework rate as DORA defines it counts deployments, and is a leading indicator of this cost.

Costs, Risks, and Tradeoffs

Exhibit 3. Tradeoffs of a monthly delivery review

Area	Cost or risk	Mitigation
Tooling	Pipelines and incident tools must record deployments and failures consistently	Start with the top five services; most CI/CD and incident tools already hold the data
Gaming	Teams split deployments to raise frequency or reclassify incidents to lower fail rate	Pair throughput with instability; audit incident classification quarterly
Morale	Engineers read any measurement as surveillance	Team-level only; publish the rules; never tie numbers to individual pay [8][9]
Leadership time	Another monthly agenda item	Fold it into the existing operating review; fifteen minutes on trends and exceptions
False comfort	Good numbers on the measured services hide unmeasured ones	Keep an inventory of critical services and their coverage
When this is wrong	A company with little in-house software, running mostly on SaaS, has few deployments to measure	Review vendor SLAs, incident counts and the change calendar for its configuration changes instead

Source: author's analysis.

How to Act

Exhibit 4. A one-year roadmap

Window	Actions	Owner	Milestone
First 30 days	Name the five services that matter most to revenue or regulation. Agree the five measure definitions in writing, using DORA's [1]. Publish the rules: per service, team level, no individual use.	CTO with CFO	Five services, written definitions, rules circulated to engineering
First 30 days	Pull the last six months of deployment and incident data for those services.	Engineering leads	Baseline for each measure, even if partial
First 90 days	Automate collection from the deployment pipeline and incident tool. Present the first monthly review.	Platform or DevOps lead	Three consecutive monthly reviews delivered from automated data
First 90 days	Set a direction and a threshold for each measure per service. Add AI tool usage as context beside delivery measures.	CTO with service owners	Targets agreed for all five services; usage shown next to outcomes
First 365 days	Extend to the next tier of services. Tie engineering investment requests to the measure they should move. Review AI tool spend against throughput and instability trends.	CTO with CFO	Every major engineering investment names its target measure; change fail rate and lead time improved on at least three of five services

Source: author's recommendation.

Questions to Ask Your Team

1. For our five most important services, how long does a finished change take to reach customers?
2. What share of last quarter's releases needed a rollback or an emergency fix?
3. When a release goes wrong, how long until customers stop feeling it?
4. How much of our engineering time last quarter went to work nobody planned?
5. Which of these numbers can we pull automatically today, and which would someone have to type in?
6. Since we adopted AI coding tools, what happened to lead time and change fail rate on the services that use them most?
7. Which of our engineering metrics could a team improve without customers noticing any difference?
8. What would we need to see in these numbers to justify the next engineering hire?

Closing

Leadership would never fund the sales organization on headcount and activity reports alone, and it should stop funding engineering that way. Five measures, reviewed monthly for each critical service, show whether changes reach customers quickly and safely, and whether new tools and new hires made that better. Agree the definitions, collect the data automatically, keep it at team level, and put it on the same agenda as the pipeline.

Sources

[1] DORA. Guide to DORA's software delivery metrics, with definitions of the throughput and instability measures and guidance on their use. 2025. https://dora.dev/guides/dora-metrics/.

[2] Forsgren, N., Humble, J., and Kim, G. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution. 2018. https://itrevolution.com/product/accelerate/.

[3] Google Cloud. "Announcing the 2024 DORA report." 2024. https://cloud.google.com/blog/products/devops-sre/announcing-the-2024-dora-report.

[4] Google Cloud. "Announcing the 2025 DORA Report: State of AI-assisted Software Development." 2025. https://cloud.google.com/blog/products/ai-machine-learning/announcing-the-2025-dora-report.

[5] DORA. "State of AI-assisted Software Development 2025." Report page. 2025. https://dora.dev/research/2025/dora-report/.

[6] Becker, J., Rush, N., Barnes, E., and Rein, D. "Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity." METR. arXiv:2507.09089. 2025. https://arxiv.org/abs/2507.09089.

[7] METR. "We are Changing our Developer Productivity Experiment Design." February 24, 2026. https://metr.org/blog/2026-02-24-uplift-update/.

[8] Forsgren, N., Storey, M.-A., Maddila, C., Zimmermann, T., Houck, B., and Butler, J. "The SPACE of Developer Productivity: There's more to it than you think." ACM Queue 19(1). 2021. https://www.microsoft.com/en-us/research/publication/the-space-of-developer-productivity-theres-more-to-it-than-you-think/.

[9] Orosz, G., and Beck, K. Response to McKinsey's proposal for measuring developer productivity. The Pragmatic Engineer. August 29, 2023. https://newsletter.pragmaticengineer.com/p/measuring-developer-productivity.

[10] GitClear. "AI Copilot Code Quality: 2025 Research." 211 million changed lines, 2020 to 2024. 2025. https://www.gitclear.com/ai_assistant_code_quality_2025_research.

[11] Sonos, Inc. Q3 fiscal 2024 earnings call transcript, via The Motley Fool. August 8, 2024. https://www.fool.com/earnings/call-transcripts/2024/08/08/sonos-sono-q3-2024-earnings-call-transcript/.

[12] Yahoo Tech. Coverage of Sonos CEO Patrick Spence stepping down. January 13, 2025. https://tech.yahoo.com/apps/articles/sonos-ceo-patrick-spence-steps-155527619.html.

[13] CNBC. Coverage of Shopify CEO Tobi Lütke's memo on AI use and headcount. April 7, 2025. https://www.cnbc.com/2025/04/07/shopify-ceo-prove-ai-cant-do-jobs-before-asking-for-more-headcount.html.

[14] CrowdStrike. "External Technical Root Cause Analysis: Channel File 291." 2024. https://www.crowdstrike.com/wp-content/uploads/2024/08/Channel-File-291-Incident-Root-Cause-Analysis-08.06.2024.pdf.

The views in this paper are the author's own and do not represent any employer. Scenarios are hypothetical. This paper is not legal, financial, or compliance advice.