

Healthy AI First, Episode 2: Where the Time Goes

Sources and evidence for this episode

When AI writes the code, the work does not disappear. It moves: from writing to explaining, checking and integrating. The episode follows one change along the whole journey from ticket to customer, shows with survey and telemetry numbers why more code is a poor measure of success, why a pile forms in front of the review bottleneck and what "almost right" costs. The tool tip shows how to read lead times from Git and GitHub instead of guessing.

All links were checked on 17 October 2026.

What we mention in this episode

Welcome

- **Claim:** In episode 1, Daniel stood for experienced developers who felt faster with AI and were measured slower.
Source: Becker, J., Rush, N., Barnes, E. & Rein, D. (METR, 2025): Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity. Preprint.
Link: <https://arxiv.org/abs/2507.09089>
Strength and limits: Randomized study, but only 16 people and tools from early 2025 (details in episode 1).

Miriam's Status Report

- **Claim:** A survey of more than three hundred knowledge workers about almost a thousand situations shows: critical thinking shifts to verifying, integrating and steering; more trust in AI goes with less critical thinking, more self-confidence with more.
Source: Lee, H.-P., Sarkar, A., Tankelevitch, L., Drosos, I., Rintel, S., Banks, R. & Wilson, N. (Microsoft Research and Carnegie Mellon, 2025): The Impact of Generative AI on Critical Thinking: Self-Reported Reductions in Cognitive Effort and Confidence Effects From a Survey of Knowledge Workers. CHI 2025.
Link: <https://doi.org/10.1145/3706598.3713778> (PDF: https://advait.org/files/lee_2025_ai_critical_thinking_survey.pdf)
Strength and limits: 319 people, 936 examples, self-report. Measures reported thinking effort, not thinking ability, and shows associations, not causes.

A Sixth of the Day

- **Claim:** Developers say they spend only about 16% of their time coding; 68% save more than ten hours a week with AI, and 50% lose about as much to organizational friction.
Source: Atlassian (2025): State of Developer Experience Report 2025.
Link: <https://www.atlassian.com/blog/developer/developer-experience-report-2025>
Strength and limits: Vendor survey of 3,500 developers and managers in six countries; self-report.
- **Claim:** At Anthropic, more than half of the respondents can fully hand over at most a fifth of their work to AI; one engineer estimates that more than 70% of his work is now reviewing; there is

concern that skills needed for supervision may wither.

Source: Anthropic (2 Dec 2025): How AI is transforming work at Anthropic.

Link: <https://www.anthropic.com/research/how-ai-is-transforming-work-at-anthropic>

Strength and limits: Internal survey of 132 people plus 53 interviews at an AI vendor; selective setting, no control comparison. The 70% is a single voice.

The Second Half of the Sentence

- **Claim:** In October 2024, Google's CEO said that more than a quarter of new code at Google is generated by AI, then reviewed and accepted by engineers.
Source: Alphabet (29 Oct 2024): Q3 earnings call, CEO remarks by Sundar Pichai.
Link: <https://blog.google/company-news/inside-google/message-ceo/alphabet-earnings-q3-2024/>
Strength and limits: Company statement, not an independent measurement.
- **Claim:** If existing project components are missing from context, an agent may overlook them and write a duplicate; each session starts with a fresh context built from user input, automatically loaded project notes, history and tool results. Searching first can reduce the risk.
Source: Anthropic: Claude Code documentation, How Claude remembers your project.
Link: <https://code.claude.com/docs/en/memory>
Strength and limits: Vendor documentation, technical explanation; that duplicate functions arise this way is a plausible mechanism, not a measured finding.
- **Claim:** Analyses of large code bases show more copied and less cleaned-up code; that AI is the cause is not proven.
Source: GitClear (2025): AI Copilot Code Quality: 2025 Data Suggests 4x Growth in Code Clones.
Link: https://www.gitclear.com/ai_assistant_code_quality_2025_research
Strength and limits: 211 million changed lines (2020 to 2024); vendor data, correlation.
- **Claim:** When a measure becomes a target, it stops being a good measure.
Source: Goodhart, C. (1975), in the well-known short form by Strathern, M. (1997): "Improving ratings": audit in the British University system. European Review 5(3), 305-321.
Link: <https://doi.org/10.1017/S1062798700002660> (overview: https://en.wikipedia.org/wiki/Goodhart%27s_law)
Strength and limits: A rule of experience from economics and anthropology, not an experiment. Covered in depth in episode 8.
- **Claim:** A better number than code volume is the time from ticket to customer.
Source: DORA: Software delivery performance metrics (including change lead time).
Link: <https://dora.dev/guides/dora-metrics/>
Strength and limits: The DORA metric runs from commit to production; "ticket to customer" is deliberately broader and a recommendation of the episode.

The Pile Before the Bottleneck

- **Claim:** Teams with high AI adoption merge almost twice as many changes (+98%), changes get much bigger (+154%), recorded review time is almost twice as high (+91%, a group-level measure, not a measurement of every individual reviewer's active work), and at company level there is no significant correlation.
Source: Faros AI (23 Jul 2025): The AI Productivity Paradox Report 2025.
Link: <https://www.faros.ai/blog/ai-software-engineering>

Strength and limits: Telemetry from more than 10,000 developers in 1,255 teams; vendor data, correlation.

- **Claim:** A system is only as fast as its narrowest point; half-finished work piles up in front of it.
Source: Goldratt, E. M. & Cox, J. (1984): The Goal. Theory of constraints.
Link: https://en.wikipedia.org/wiki/Theory_of_constraints (overview)
Strength and limits: Management concept from manufacturing; applying it to software review is a thinking model.

Almost Right

- **Claim:** Long, focused checking is tiring; you are more likely to miss something.
Source: Warm, J. S., Parasuraman, R. & Matthews, G. (2008): Vigilance Requires Hard Mental Work and Is Stressful. Human Factors 50(3), 433-441.
Link: <https://doi.org/10.1518/001872008X312152>
Strength and limits: Review of vigilance research; transferred to code review, not measured there.
- **Claim:** In one of the largest developer surveys, about two thirds name "almost right, but not quite" as their biggest frustration with AI.
Source: Stack Overflow (2025): Developer Survey 2025, AI section.
Link: <https://survey.stackoverflow.co/2025/ai>
Strength and limits: 66% of those answering this question; large but self-selected sample.
- **Claim:** The most frequent reason for confusion in code reviews is a missing rationale for the change.
Source: Ebert, F., Castor, F., Novielli, N. & Serebrenik, A. (2019): Confusion in Code Reviews: Reasons, Impacts, and Coping Strategies. SANER 2019.
Link: <https://doi.org/10.1109/SANER.2019.8668024>
Strength and limits: Survey plus analysis of review comments; qualitative evidence. The "three sentences on top of every change request" are a recommendation derived from it.

The Clock in the Tools

- **Claim:** Git stores a timestamp with every commit, the GitHub CLI shows per pull request when it was opened, reviewed and merged, and a coding agent can call both itself.
Source: Git, GitHub CLI and Claude Code documentation (see tool tip).
Strength and limits: Official documentation. The timestamps give calendar intervals (lead times); they do not separate active work, waiting, nights and rework, and a submitted review is not when reading began.

Your Time, Split Up

- **Claim:** A document's version history often shows who worked on it and when.
Source: Google Docs Editors Help: Find what's changed in a file.
Link: <https://support.google.com/docs/answer/190843>
Strength and limits: Vendor documentation; not every program offers this.

Tool tip: read your lead times instead of guessing

Example prompt: "For my last 10 pull requests, show me the elapsed time between the first commit, opening, first submitted review and merge." These are lead-time intervals, not pure waiting times. You need additional records to separate active work from actual waiting.

- `gh pr list --author @me --state merged --limit 10 --json number,title,createdAt,mergeDate,reviews,commits`: creation and merge time, reviews and commits per pull request. https://cli.github.com/manual/gh_pr_list
- `git log --format='%h %aI %s'`: commits with their author date; a first commit is not necessarily when work started. <https://git-scm.com/docs/git-log>
- Claude Code with `gh` and in scripts (`git log | claude -p "..."`): <https://code.claude.com/docs/en/common-workflows>
- A notes file against duplicate functions: CLAUDE.md (loaded in every session) <https://code.claude.com/docs/en/memory> · the open AGENTS.md format <https://agents.md>
- Terms pull request, diff, merge: <https://docs.github.com/en/pull-requests/reference/pull-requests>

Further reading

- Baolin, Harvey (DORA, 10 Mar 2026): Balancing AI tensions. Thematic analysis of 1,110 open answers from Google engineers; saved time is redistributed into review and prompting effort. Qualitative, one company. <https://dora.dev/insights/balancing-ai-tensions/>
- DORA / Google Cloud (2024): Announcing the 2024 DORA report. More AI adoption goes with slightly lower delivery throughput and stability (survey, correlation). <https://cloud.google.com/blog/products/devops-sre/announcing-the-2024-dora-report>
- Dijkstra, E. W. (1988): EWD 1036, On the cruelty of really teaching computing science ("lines spent" rather than "lines produced"). <https://www.cs.utexas.edu/~EWD/transcriptions/EWD10xx/EWD1036.html>
- Basis of the series: Healthy AI First Software Development, version 2 (1 October 2026), chapter 3.

Note

Healthy AI First is AI-researched and AI-generated, narrated in Björn's voice, recreated with AI, and reviewed by him.

Jonas, Daniel, Miriam and Hanna are fictional characters used as illustrations; their scenes (dashboard, fourteen pull requests, status report, wedding photos) are invented.