JUST TECH

Cognitive Debt and AI Vampires: Generative AI and the New Software Engineering Workplace

By: Bhargav Srinivasa Desikan

LABOR & ECONOMY

Desikan, Bhargav Srinivasa. "Cognitive Debt and AI Vampires: Generative AI and the New Software Engineering Workplace." Just Tech. Social Science Research Council. November 4, 2026. DOI: https://doi.org/10.35650/JT.3106.d.2026.

"I am getting more time, but I am exhausted."[1] That's what Simon Willison, a programmer with 25 years of experience and one of the cocreators of the Django web framework, told *Business Insider* about working with generative AI.[2] Willison is as senior and independent as a software engineer can be, choosing his own hours and his own projects. With AI, he can run four agents in parallel on four different problems but is worn out by around 11 in the morning. The tools meant to save him time have made his days far more stressful.

At the other end of the profession, the same tools are described differently. A junior engineer at a large San Francisco technology company told the *San Francisco Standard* that he had become "basically a proxy to Claude Code," with his manager giving him instructions that he passes on to the AI. He said he mostly felt grief that a skill he had taken years to build was now available to anyone, and he worried that any new model release might be the one that made his job unnecessary.[3]

Popular accounts of generative AI and coding tell two stories that seem to pull in opposite directions, though both can be true at once. On one side, the dream of *vibe coding*, a term coined by OpenAI founding member Andrej Karpathy to describe building software through natural-language prompts while you "forget that the code even exists."[4] On the other hand, we have constant news of AI-driven layoffs, with nearly every major tech CEO announcing AI-related restructurings.

Despite it seeming an insular preoccupation, there is a reason to think through this situation: Software

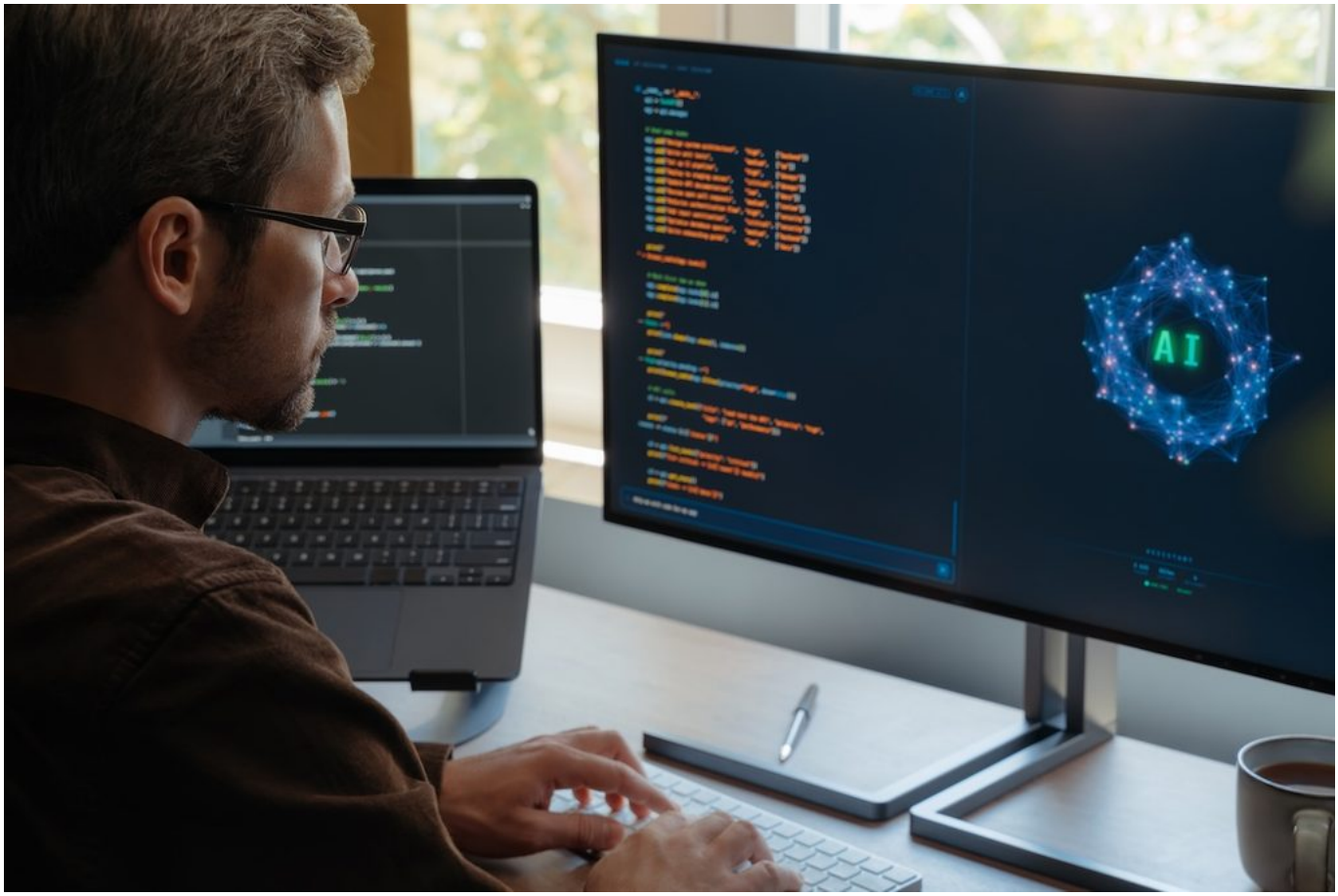
engineering is arguably the first skilled profession in which generative AI has become part of the everyday infrastructure of work. In 2025, 84 percent of developers surveyed by Stack Overflow said they were using or planning to use AI tools, and by mid-2026 nine in ten developers surveyed by JetBrains were using AI coding agents at least once a week.[5] Software engineers' future may foretell the fate of other kinds of knowledge work. However, the accounts of this change that circulate most widely tend to be written from the top of the profession, by the firms selling the tools and the engineers who have the most to gain. If we mistake their view for a general one, we will misread what AI is doing to the people further down, the junior developers and maintainers whose work is the easiest to hand over to an agent, and the offshore engineers in cities like Bangalore, whose place in the global economy was built on demand for cheap code.

Where an engineer sits in the labor hierarchy shapes how generative AI impacts their work. This shows up most clearly in two places: in engineers' understanding of the code they are responsible for, which many of them now describe as *cognitive debt*, and in their working time, where the question is who keeps the hours the tools save, a problem the programmer Steve Yegge describes as the *AI vampire*.[6] To make this case, I draw on what engineers themselves have written about AI over the past two years, and on patterns across more than four million posts and comments on Hacker News and Reddit, two forums where software engineers discuss their work at length.[7] I also draw on conversations with friends and former colleagues in the tech industry, where I worked before moving into research.

Skill

Software engineers have begun naming what is happening to their understanding of their own code. The software engineering researcher Margaret-Anne Storey described a student team on a course she taught who, seven or eight weeks into building a product, could no longer make simple changes without breaking something, and that nobody could explain design decisions. Storey uses a term now common in software engineering, *cognitive debt*, for the gap that opens between a system and a team's understanding of it: "Technical debt lives in the code. Cognitive debt lives in people."[8] The developer Namanyay Goel described realizing, during a ChatGPT outage, that he had stopped reading documentation and had fallen into the habit of pasting error messages into a chatbot instead of working out what they meant.[9]

Not everyone sees this as a loss. For some senior developers, handing over the typing is the point; one wrote that working with AI made him feel more like a creative director than an assembly-line worker.[10] Some employers are also keen. The engineer Chris Dzombak has argued that many companies are comfortable encouraging AI-assisted coding because they value shipping quickly over deep familiarity with a codebase, moving engineers between projects before they can build real expertise in any of them. The tools fit a way of working that already existed in the industry.[11]



One software developer explained how AI coding tools "freed them to be a better engineer." Photo source: Adobe Stock.

This is an old argument in the sociology of work. Harry Braverman showed how capitalist technological change tends to separate the conception of work from its execution, shifting the authority of knowledge from workers to management and machines.[12] Software engineering has long been held up as a counterexample, a craft whose practitioners do both the thinking and the doing, though this has always been contested. Studies of software firms have found managers alternating between granting engineers autonomy and tightening control depending on a project's urgency.[13]

The arrival of AI coding tools reopens this debate. If developers can produce code by prompting an AI system rather than writing it themselves, the execution of the task is still there, only now it is mediated through the model. The developer may still know how to code, but they are no longer working through it line by line. Read through *labor process theory*, the cognitive debt engineers describe looks like a new turn in Braverman's story: The engineer still carries out the task and is responsible for the result, but the understanding that used to come from doing the work is gone, and code they may not be able to fully explain, debug, or maintain still goes out under their name and authorship. Vibe coding makes this explicit: It promises no longer needing to understand the code you write.

The human factors literature has long identified this pattern in other domains, and Lisanne Bainbridge's classic account of the "ironies of automation" showed how, as automated systems become more reliable, their human operators are reduced to monitoring them and lose the vigilance needed to step in when

they fail.[14] The additional twist in software is that the developer keeps a sense of freedom (choosing what to prompt and what to accept), but even that may not last, as expertise wears away just as workplace expectations rise.

The same concern shows up online, across more than four million posts and comments on Hacker News and Reddit. When developers describe what AI has made faster, they overwhelmingly describe execution (writing code, debugging, generating documentation) and far less often design, architecture, or strategy, by a ratio of roughly six or seven to one.[15] Since these tools arrived, talk of skill loss has also grown faster in work discussions that mention AI than in those that don't. It concentrates on technical debt (the accumulated cost of past shortcuts in a codebase), code review, and debugging, the activities where understanding a codebase matters most.[16]

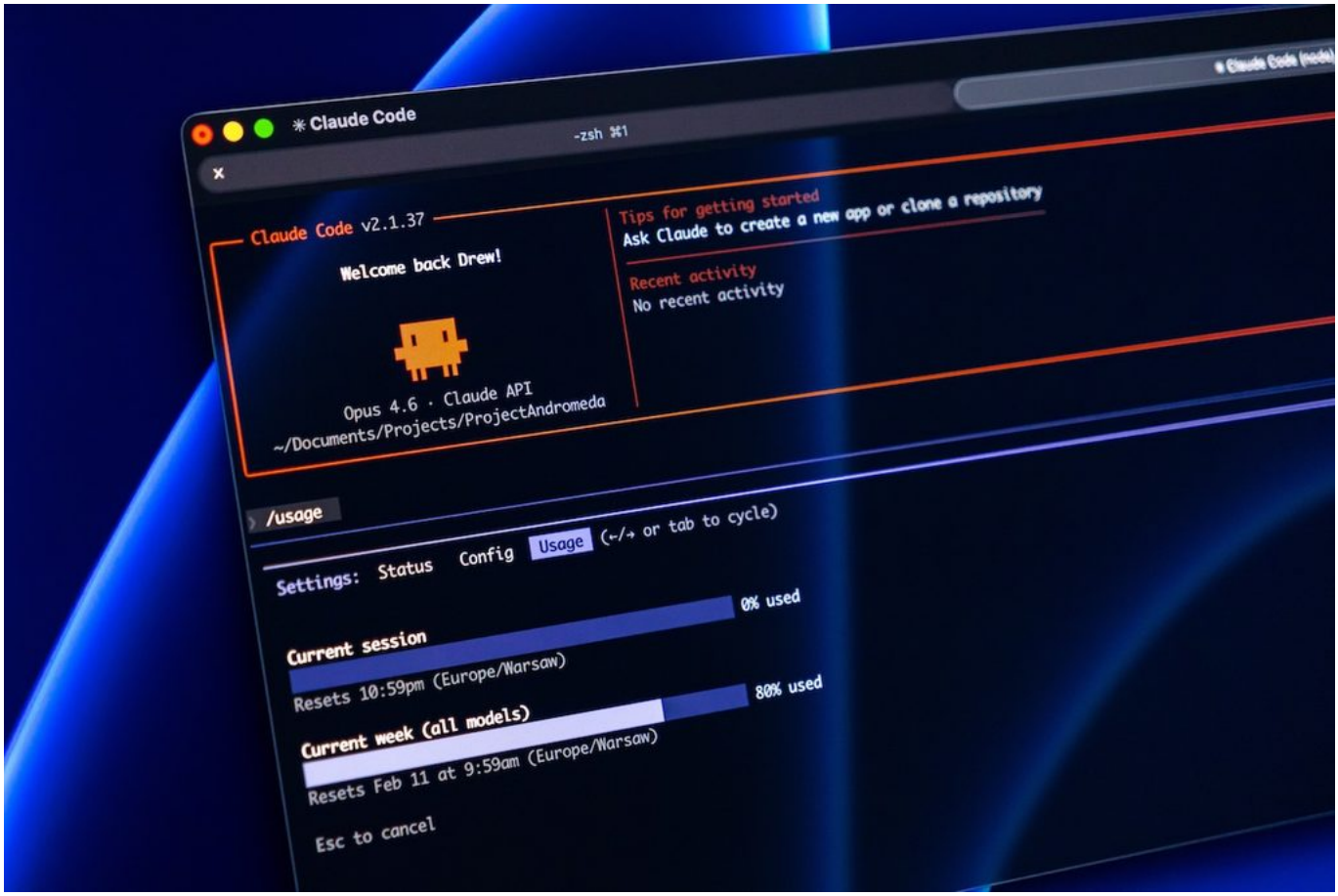
Unsurprisingly, the stakes of skill loss differ by position. For a chief technology officer, these tools look like good news for the business: More gets shipped with fewer people, and decisions about what to build sit even more firmly with those at the top. For an engineer with little power, the same tools drain away the parts of the job that were theirs—the problem-solving and the small decisions that made the work satisfying—and leave them passing instructions to a model and checking what comes back. What they had to sell, the ability to write and understand code, is also worth less. As a result, more of the strategic thinking moves up the hierarchy, and engineers have less to bargain with.

The evidence on who benefits in the labor hierarchy is mixed; field experiments at Microsoft, Accenture, and a Fortune 100 company found that less experienced developers gained the most from an AI coding assistant.[17] And yet, in the United States, employment of 22- to 25-year-olds in the occupations most exposed to AI now stands 19 percent below where it would have been had it kept pace with their less exposed peers, largely because fewer of them are being hired.[18]

Time

As with skill, the most vivid accounts of time under generative AI come from engineers themselves. Yegge, a veteran programmer, likens AI to an energy vampire: The tools make engineers far more productive, and if their employer captures all of that extra output, the engineer is simply drained. He describes sudden "nap attacks" after long sessions with coding agents and argues that the working day for this kind of work should be three or four hours.[19] Quentin Rousseau compared the pull of agentic coding to a slot machine: You never know whether the next prompt will produce something brilliant or useless, so you keep pulling.[20] Researchers are finding something similar. An eight-month study of a US technology firm found that AI intensified work, as employees took on more and let it spill into what had been pauses in the day, often without anyone asking them to.[21] Researchers at the Boston Consulting Group and the University of California, Riverside, coined the term *AI brain fry* for the mental fatigue that comes from overseeing AI tools beyond a person's cognitive capacity. They found it concentrated among the heaviest users and those running several agents at once.[22]

Willison has said that 25 years of intuition about how long a task will take are no longer useful, because something he would once have judged a fortnight's work might now take 20 minutes. So, he gives the agents tasks he doubts they can do, since trying costs him little and every so often they manage it.[23] Eric Dodds, who works in the industry and uses these tools heavily, wrote that the natural variety of a working week used to give his brain regular breaks. Now that Claude is connected to the tools his team plans and talks in, he has to insist on breaks, because it is possible to work at full stretch on hard problems around the clock.[24] Even the shape of the day is starting to be set by the tools, as developers talk about organizing their work around the five-hour resets of their usage limits.[25]



Due to Claude's five-hour limit, users have to use tricks to circumvent them. Photo source: Adobe Stock.

The sociology and anthropology of time help make sense of this. Judy Wajcman argues that the feeling of acceleration in digital capitalism is produced by the institutional and labor-market arrangements through which technologies are taken up, which is why the same tool can free one worker's time and tighten another's.[26] She also draws on what Mazmanian, Orlikowski, and Yates call the "autonomy paradox," in which professionals experience constant connectivity as flexibility and control, even as it erodes their autonomy.[27] Sarah Sharma takes this further with her concept of power-chronography: Time is experienced hierarchically and in relation to others, and the smooth, uninterrupted time of those at the top depends on others whose time is made to bend around it. Her term for the work that less powerful people do to fit themselves to dominant rhythms is *recalibration*.[28]

For some engineers, AI tools free time. The senior architect at a well-resourced firm, who already controlled the shape of their working day, can compress execution and potentially reclaim hours for design, mentorship, or simply leaving the office at a reasonable hour. They keep the hours they save, though, as Willison's case shows, the days can still leave them drained.

For others, the same tools produce the opposite experience. Junior developers, engineers at firms with tight delivery cycles, and offshore workers whose targets keep being reset against what the leading firms achieve find that AI increases output expectations within the same number of hours. For them, Yegge's question of who captures the value has a clear answer: The time AI saves goes into tighter deadlines, bigger scope, and higher targets, and little of it comes back to the worker.

Online developer discussions show this happening over time. Reddit posts describe a brief window in 2023 and early 2024, before managers had fully registered what the tools could do, in which engineers could quietly absorb some of the time they saved into their own working day. From late 2024 onward, they describe managers rebudgeting deadlines downward as those gains are folded into output expectations. Developers also describe the gains differently depending on how much control their community has over its own work. On r/webdev, a Reddit community of web developers that leans toward freelancing and tool enthusiasm, framings of AI as liberating outnumber framings of it as extractive by something like six to one, whereas on r/cscareerquestions, a community of students, job-seekers, and early-career developers, the two are close to parity. This fits Sharma's account, though forum posts show how people talk about their time, which may differ from how they live it.

Bangalore

Nowhere are these questions more pressing than in Bangalore, the center of gravity of India's technology industry, which employs close to six million people.[29] In July 2025, Tata Consultancy Services (TCS), India's largest information technology services company, announced it would cut around 12,000 jobs, about 2 percent of its workforce, mostly in middle and senior grades. The company put the cuts down to a mismatch of skills, but reporting framed them as the start of an AI-driven shift that could remove around half a million jobs from India's $283 billion outsourcing sector within two or three years.[30] Bangalore's technology economy was built on demand for skilled software labor that was cheaper than its US and European equivalents, and AI coding tools are being sold, among other things, as a way to make that labor even cheaper.

Indian software engineers have already been studied as a distinctive labor formation. In Bangalore, the industry drew on and expanded India's English-educated middle class, with its profits resting largely on the difference in labor costs.[31] Abroad, Indian engineers circulated through systems of contract "body shopping,"[32] and Sareeta Amrute, working with Indian programmers in Berlin, has shown coding itself to be an embodied, racialized, and classed practice.[33] Lilly Irani, following a Delhi design studio, traces how designers, businesspeople, and state policy have tried to make Indians into entrepreneurial citizens, and Hemangini Gupta's ethnography of Bangalore's start-up scene shows that the freedom to experiment, and

the time it takes, rests on class and gender privilege.[34] Generative AI now unsettles the conditions all of this research describes.

Many of the friends I studied computer science with now work in Bangalore's technology firms, and in conversations with them it has struck me how differently they're experiencing this moment. Some are exhausted by toxic workplaces, some are nostalgic for slower ways of working, and some are genuinely thrilled by what AI promises. A senior figure in the city's AI and software-as-a-service founder community voiced a different anxiety to me: She wondered why India had not taken off with AI in the way it had with software-as-a-service and suspected that the conditions that made Indian software companies successful might not carry over. The anthropologist Laura Bear writes of labor "in and of time," the work of reconciling competing rhythms.[35] In Bangalore it is part of daily life, with the working day stretched between Silicon Valley product cycles, global delivery schedules, and late calls synchronized to other people's working hours.

These pressures will sound familiar, but in Bangalore they fall on already long working days. In 2024, the Karnataka government considered allowing IT firms to stretch the day to 14 hours and dropped the plan only after protests led by the Karnataka State IT/ITeS Employees Union (KITU).[36] For Bangalore's engineers, AI adds a double bind. In the immediate term, productivity gains translate into tighter deadlines and expanded scope, and in the longer term, the very tools and ecosystem they are helping build risk automating away the cost advantage that sustains their position in the global labor market.[37] On top of everything else, the voices of Indian developers are seldom heard in public discourse on AI's impact. Most of the essays and posts I have drawn on were written by engineers in the United States and Europe. Overall, these divisions are arguably the sharpest in Bangalore, and it is also where they are the least documented.



Google workers walk out in protest of the company's contracts in 2018. Photo source: Grendelkhan/Wikimedia Commons.

Consent and Class Consciousness

Why do engineers go along with the use of generative AI? Here Michael Burawoy's *Manufacturing Consent* is useful.[38] Working as a machine operator in a South Chicago factory in the mid-1970s, Burawoy found that workers' cooperation came from informal shop-floor games. The main one was "making out," beating the piece rate to earn a bonus, which made the work absorbing while raising output for the firm. Arguably, the loop of prompting, generating, reviewing, and shipping is a similar game, and the slot-machine pull Rousseau describes, together with agents that keep offering to do one more thing, makes it a very absorbing one. Talk of skill loss online is mostly skeptical or ironic, and few people admit to feeling deskilled themselves, even as the comprehension problem grows. In Burawoy's terms, this is consent to the game: Engineers can see what it may cost them and keep playing. In one randomized trial, experienced open-source developers were 19 percent slower when using AI tools even though they believed the tools had made them faster,[39] a reminder that the rush of seeming to get things done may be its own reward. At the same time, the game is not entirely voluntary. A Hacker News commenter complained that engineers are now expected to perform excitement about AI several times a day, and METR, the research nonprofit behind the AI tool trial, reported that by late 2025 many of the developers it recruited were unwilling to do some of their tasks without AI at all.[40]

And yet consent is not the whole story. Among my colleagues, friends, and fellow researchers, I see a growing consciousness of the battles over power and control in the tech industry. Software engineers

have long been treated as the comfortable middle class of the knowledge economy, and yet in recent years they have formed unions, such as the Alphabet Workers Union at Google, and organized campaigns, such as No Tech for Apartheid, against their own employers' contracts. Generative AI makes these conversations more personal, because what is at stake now is the value of engineers' own labor. Overall, consent and a growing awareness of how the gains are shared sit side by side in this discourse, often in the same people.

In Bangalore, too, some engineers are organizing. KITU, the union behind the protests against the 14-hour day, filed an industrial dispute against TCS within days of the layoff announcement.[41] Offshoring arguably set engineers in San Francisco and Bangalore against each other, competing on the price of the same work. AI now puts both under similar pressure on skill and time, even if engineers in Bangalore feel it more acutely. Whether that becomes a basis for organizing across borders is, to my mind, one of the more important open questions AI raises for work.

Footnotes

1 Ben Shimkus, "'AI-pilled' Engineers Are Working Harder and Burning Out Faster, Django Cocreator Says," *Business Insider*, April 3, 2026, https://www.businessinsider.com/ai-engineers-exhausted-django-co-creator-simon-willison-2026-4.

2 Simon Willison, "Highlights from my Conversation about Agentic Engineering on Lenny's Podcast," Simon Willison's Weblog, April 2, 2026, https://simonwillison.net/2026/Apr/2/lennys-podcast/.

3 Rya Jetha, "AI Writes the Code Now. What's Left for Software Engineers?," *San Francisco Standard*, February 19, 2026, https://sfstandard.com/2026/02/19/ai-writes-code-now-s-left-software-engineers/.

4 Andrej Karpathy (@karpathy), "There's a new kind of coding I call 'vibe coding,' where you fully give in to the vibes, embrace exponentials, and forget that the code even exists...," Twitter (now X), February 2, 2025, https://x.com/karpathy/status/1886192184808149383.

5 Stack Overflow, "AI Tools in the Development Process," 2025 Developer Survey, https://survey.stackoverflow.co/2025/ai#1-ai-tools-in-the-development-process, and Mikhail Bogdanov, "AI Coding Agents: Adoption Trends," *JetBrains Blog*, August 18, 2026, https://blog.jetbrains.com/research/2026/08/ai-coding-agent-adoption-2026/.

6 Steve Yegge, "The AI Vampire," Medium, February 11, 2026, https://steve-yegge.medium.com/the-ai-vampire-eda6e4f07163.

7 What the online blog and microblog landscape offers is a map of these divisions, drawn by the people who post, and it has its limits. Hacker News, for example, leans toward founders and senior engineers.

8 The student team appears in Margaret-Anne Storey, "How Generative and Agentic AI Shift Concern from Technical Debt to Cognitive Debt," margaretstorey.com, February 9, 2026, https://margaretstorey.com/blog/2026/02/09/cognitive-debt/; the quoted lines are from her follow-up, "What I'm Hearing about Cognitive Debt (So Far)," margaretstorey.com, February 18, 2026, https://margaretstorey.com/blog/2026/02/18/cognitive-debt-revisited/.

9 Namanyay Goel, "AI Is Creating a Generation of Illiterate Programmers," nmn.gl, January 24, 2025, https://nmn.gl/blog/ai-illiterate-programmers.

10 Shabbir Hussain, "I Stopped Googling Code in 2026 — Here's How AI 'Vibe Coding' Changed Everything," Medium, January 20, 2026, https://medium.com/@001.shabbirhussain/i-stopped-googling-code-in-2026-heres-how-ai-vibe-coding-changed-everything-1ffe93988c63.

11 Chris Dzombak, "Why Tech Companies Encourage Their Software Engineers to Use LLMs," dzombak.com, December 18, 2025, https://www.dzombak.com/blog/2025/12/why-tech-companies-encourage-their-software-engineers-to-use-llms/.

12 Harry Braverman, *Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century* (New York: Monthly Review Press, 1974).

13 Braverman's account, powerful as it was, invited correction. Andrew Friedman distinguished direct control from "responsible autonomy," in which managers grant workers bounded independence as itself a strategy of control, and Richard Edwards set out a typology of simple, technical, and bureaucratic control. In Friedman's account the two strategies are applied to different groups of workers, while Rowena Barrett, in one of the few studies to apply labor process theory directly to software engineers, found them coexisting in a single firm. Yigang Qin and EunJeong Cheon have recently argued that labor process theory is an underused resource for research on human-computer interaction. See Andrew L. Friedman, *Industry and Labour: Class Struggle at Work and Monopoly Capitalism* (London: Macmillan, 1977); Richard Edwards, *Contested Terrain: The Transformation of the Workplace in the Twentieth Century* (New York: Basic Books, 1979); Rowena Barrett, "Working at Webboyz: An Analysis of Control over the Software Development Labour Process," *Sociology* 38, no. 4 (2004): 777–794, https://doi.org/10.1177/0038038504045864; and Yigang Qin and EunJeong Cheon, "Labor, Capital, and Machine: Toward a Labor Process Theory for HCI," in *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems* (New York: ACM, 2026), https://doi.org/10.1145/3772318.3791147. On the routinization of programming, see also Philip Kraft, *Programmers and Managers: The Routinization of Computer Programming in the United States* (New York: Springer-Verlag, 1977).

14 Lisanne Bainbridge, "Ironies of Automation," *Automatica* 19, no. 6 (1983): 775–779, https://doi.org/10.1016/0005-1098(83)90046-8.

15 These results are from unpublished analysis for my DPhil thesis. Economists studying open-source projects have found a similar asymmetry: GitHub Copilot increased contributions that maintain and refine existing code much more than those that add new functionality. See Doron Yeverechyahu, Raveesh Mayya, and Gal Oestreicher-Singer, "The Impact of Large Language Models on Open-Source Innovation: Evidence from GitHub Copilot," preprint, arXiv, revised May 24, 2026, https://arxiv.org/abs/2409.08379.

16 From the same analysis: comparing work discussions that do and do not mention AI, before and after the 2022–2023 adoption wave, shows a small-to-moderate rise in deskilling language (Cohen's d = 0.30), with no effect at a placebo date set before these tools existed. Because developers choose whether to mention AI, this is best read as an association.

17 Kevin Zheyuan Cui et al., "The Effects of Generative AI on High-Skilled Work: Evidence from Three Field Experiments with Software Developers," *Management Science*, February 27, 2026, https://doi.org/10.1287/mnsc.2025.00535.

18 Erik Brynjolfsson, Bharat Chandar, and Ruyu Chen, "Canaries in the Coal Mine? Six Facts about the Recent Employment Effects of Artificial Intelligence," working paper (Stanford Digital Economy Lab, Stanford University, August 12, 2026), https://digitaleconomy.stanford.edu/publications/canaries-in-the-coal-mine/.

19 Yegge, "The AI Vampire."

20 Rousseau, a chief technology officer, describes the mechanism as variable ratio reinforcement, and writes about the 2 a.m. sessions it produced for him, and about eventually seeking medical help to sleep. He credits the observation to Yegge. Quentin Rousseau, "One More Prompt: The Dopamine Trap of Agentic Coding," blog.quent.in, March 9, 2026, https://blog.quent.in/blog/2026/03/09/one-more-prompt-the-dopamine-trap-of-agentic-coding/.

21 Aruna Ranganathan and Xingqi Maggie Ye, "AI Doesn't Reduce Work—It Intensifies It," *Harvard Business Review*, February 9, 2026, https://hbr.org/2026/02/ai-doesnt-reduce-work-it-intensifies-it.

22 Julie Bedard et al., "When Using AI Leads to 'Brain Fry'," *Harvard Business Review*, March 5, 2026, https://hbr.org/2026/03/when-using-ai-leads-to-brain-fry.

23 Willison, "Highlights from my Conversation."

24 Eric Dodds, "The AI Burnout Is Real," ericdodds.com, April 15, 2026, https://www.ericdodds.com/blog/the-ai-burnout-is-real. On the review burden more generally, see Ryan Donovan, "Coding Agents Are Giving Everyone Decision Fatigue," *Stack Overflow Blog*, May 21, 2026, https://stackoverflow.blog/2026/05/21/coding-agents-are-giving-everyone-decision-fatigue/.

25 Anthropic's Claude Code meters usage through a rolling five-hour window alongside a weekly cap, and community documentation advises planning intensive work around the reset cycle, see ClaudeLog, "Claude Code Limit," last updated October 24, 2026, https://claudelog.com/faqs/claude-code-limit/. One developer describes scheduling a trivial prompt early in the morning so that the window opens at seven, turning two useful resets in the working day into three, see Teruo Kunihiro, "How I Make Claude Code's 5-hour Usage Window Last Longer on Claude Pro," dev.to, May 27, 2026, https://dev.to/trknhr/how-i-make-claude-codes-5-hour-usage-window-last-longer-on-claude-pro-jkb; and Mahnoor Faisal, "I Discovered a Hidden Trick to Reset My Claude 5-hour Window Whenever I Want…Sort of," *XDA*, June 29, 2026, https://www.xda-developers.com/discovered-a-hidden-trick-to-reset-my-claude-5-hour-window-whenever-i-want/.

26 Judy Wajcman, *Pressed for Time: The Acceleration of Life in Digital Capitalism* (Chicago: The University of Chicago Press, 2014).

27 Melissa Mazmanian, Wanda J. Orlikowski, and JoAnne Yates, "The Autonomy Paradox: The Implications of Mobile Email Devices for Knowledge Professionals," *Organization Science* 24, no. 5 (2013): 1337–1357, https://doi.org/10.1287/orsc.1120.0806.

28 Sarah Sharma, *In the Meantime: Temporality and Cultural Politics* (Durham, NC: Duke University Press, 2014).

29 The figure is from NASSCOM's Strategic Review 2026, which projects direct employment in India's technology sector at around six million in the 2026 financial year and puts the workforce at 5.8 million a year earlier, *Steering Through the Tides of Uncertainty: Charting Opportunities Amid Disruption* (NASSCOM, 2026), https://community.nasscom.in/communities/nasscom-insights/technology-sector-india-strategic-review-2026.

30 Haripriya Suresh, "Indian Tech Company TCS to Cut Workforce by 2%, Affecting More than 12,000 Jobs," Reuters, July 27, 2025, https://finance.yahoo.com/news/indian-tech-company-tcs-cut-103558769.html; and Sai Ishwarbharath B and Haripriya Suresh, "India Tech Giant TCS Layoffs Herald AI Shakeup of $283 Billion Outsourcing Sector," Reuters, August 8, 2025, https://tech.yahoo.com/ai/articles/analysis-india-tech-giant-tcs-052643308.html. The company gave the figure as roughly 12,200 jobs from a workforce of more than 613,000.

31 Carol Upadhya, *Reengineering India: Work, Capital, and Class in an Offshore Economy* (New Delhi: Oxford University Press, 2016).

32 Biao Xiang, *Global "Body Shopping": An Indian Labor System in the Information Technology Industry* (Princeton, NJ: Princeton University Press, 2007).

33 Sareeta Amrute, *Encoding Race, Encoding Class: Indian IT Workers in Berlin* (Durham, NC: Duke University Press, 2016).

34 Lilly Irani, *Chasing Innovation: Making Entrepreneurial Citizens in Modern India* (Princeton, NJ: Princeton University Press, 2019); and Hemangini Gupta, *Experimental Times: Startup Capitalism and Feminist Futures in India* (Oakland, CA: University of California Press, 2024).

35 Laura Bear, "Doubt, Conflict, Mediation: The Anthropology of Modern Time," *Journal of the Royal Anthropological Institute* 20, no. S1 (2014): 3–30, https://doi.org/10.1111/1467-9655.12091.

36 Richie Varghese Koshy, "'We Are Workers, Not Your Slaves': IT Workers in Bengaluru Protest against 14 Hour Work," *The News Minute*, August 4, 2024, https://www.thenewsminute.com/karnataka/we-are-workers-not-your-slaves-it-workers-in-bengaluru-protest-against-14-hour-work; and Korah Abraham, "Karnataka Proposes Longer Workday and More Overtime, Unions Call It 'Slavery'," *The News Minute*, June 18, 2025, https://www.thenewsminute.com/karnataka/karnataka-proposes-longer-workday-and-more-overtime-unions-call-it-slavery/.

37 Many are also contributors to "ghost work," the hidden human labor of data labelling, quality assurance, and model evaluation that sustains AI systems behind the scenes, and that is rendered invisible by the same productivity narratives that celebrate AI's capabilities. See Mary L. Gray and Siddharth Suri, *Ghost Work: How to Stop Silicon Valley from Building a New Global Underclass* (Boston: Houghton Mifflin Harcourt, 2019).

38 Michael Burawoy, *Manufacturing Consent: Changes in the Labor Process Under Monopoly Capitalism* (Chicago: The University of Chicago Press, 1979).

39 Joel Becker et al., "Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity," preprint, arXiv, July 25, 2025, https://arxiv.org/abs/2507.09089.

40 Hacker News, April 1, 2026. Posts from Hacker News and Reddit are paraphrased and not quoted to preserve authors' anonymity. On dependency, see METR's early-2026 update to Becker et al., in which 30–50 percent of participants reported withholding tasks they did not want to do without AI, reported in Julie Bort, "Coders Are Refusing to Work without AI," *TechCrunch*, May 29, 2026, https://techcrunch.com/2026/05/29/coders-are-refusing-to-work-without-ai-and-that-could-come-back-to-bite-them/.

41 C P Balasubramanyam, "Karnataka Labour Department Calls for Meeting with TCS on Aug 6 Over Layoffs," *Analytics India Magazine*, July 31, 2025, https://analyticsindiamag.com/ai-news-updates/karnataka-labour-department-calls-for-meeting-with-tcs-on-aug-6-over-layoffs.