

Life Inside the Curve

Johan Stromquist | March 2026

I. Six Sessions

Last week I had a problem. I had built six standalone applications designed to work together as a pipeline. They were not integrated. They did not share contracts. They could not speak to each other. Even with AI assistance, updating each application individually to meet a common standard would have taken days. Testing the end-to-end flow – days more.

I did not do that. Instead, I spent two hours building a lightweight coordination service – CCRouter – that sits on my local machine and exposes an MCP server that Claude Code sessions can subscribe to. CCRouter allows one Claude session to read the conversation of another, and – critically – to send messages to another session’s terminal.

I opened seven terminal windows. Each ran a Claude Code session – six pointed at the six applications, one acting as orchestrator with the full specification. Using CCRouter, the supervisor read the state of each application, identified the integration gaps, and sent application-specific instructions to each of the six worker sessions. I watched them work.

Within a few hours, all six applications were synchronised – speaking the same language, sharing the same contracts, passing tasks between each other

as the pipeline intended. A few hours after that, the supervisor had orchestrated ten full end-to-end test cycles, identified failures, dispatched fixes to the relevant sessions, and re-tested. By the end of the day, the pipeline worked.

The problem had shifted. It was no longer “how do I get these applications to talk to each other?” It was “how do I get my agents to collaborate to make this work?” And the second problem was smaller, faster to solve, and more interesting than the first.

I have been working with generative AI tools daily since January 2023 – practically since ChatGPT was released. Three years. I have watched the capabilities grow. I have written about the implications. But watching six AI sessions work in parallel – not generating content, not posting to social media, but doing actual engineering work with testable outcomes and real consequences – was something different. It was not incrementally better. It was a different kind of experience. The word that keeps coming back is *vertiginous*.

That vertigo is what this essay is about. Not the technology itself – there are benchmarks for that. Not the economic implications – I have addressed those elsewhere.¹ This essay is about what it feels like to be inside an exponential curve, why that feeling matters, and what the trajectory tells us about where we are headed.

II. The Two-Week Arc

On February 10, 2026, Matt Shumer published an essay titled “Something Big Is Happening.” It went viral in a way that AI commentary rarely does – the view count reached into the tens of millions within days. The essay’s central claim was simple: Shumer described software in plain English, walked away, and came back to find finished, correct work.² He compared

the feeling to early COVID – the phase where something enormous was clearly happening but most people had not yet adjusted their priors.

The essay was not technically novel. The AI-adjacent world had been living with this reality since Claude Opus 4.5 in December 2025 and the simultaneous release of Opus 4.6 and GPT-5.3 Codex on February 5, 2026. What made it spread was the register: it was written for family, for friends, for anyone still asking “so what’s the deal with AI?” It was a practitioner’s account of a threshold crossed, in language that non-practitioners could absorb.

The discourse it triggered was more interesting than the essay itself. Isaac Saul argued that software is a structured, pattern-rich domain, and that engineers overestimate how well AI capability transfers to messier human work – journalism, law, relationship-dependent professions.³ A writer using the handle Wilmanitas published a piece called “Tool-Shaped Objects,” arguing that LLMs produce the *feeling* of work rather than work itself – persuasive outputs that collapse under scrutiny.⁴ Connor Boyack invoked Bastiat’s distinction between the seen and the unseen: displaced jobs are visible and emotional; new industries, lowered barriers, and unlocked creativity are invisible by definition. The historical pattern – knitting machine, power loom, computer – always resolves the same way.⁵

Two weeks later, on February 20, METR – Model Evaluation & Threat Research, an organisation tracking autonomous AI capability on real-world software engineering tasks – released updated results for what has been called the most important chart in the economy: their “Moore’s Law for AI Agents.”⁶ The numbers were dramatic:

Opus 4.5, tested in December 2025, could work autonomously on a software engineering task for 4 hours and 49 minutes before the quality of its output degraded below a human baseline. Opus 4.6, tested in February 2026, reached approximately 14 hours and 30 minutes – tripling its predecessor’s result in a single generational jump.⁶

The implied doubling rate – the time it takes for the autonomous work horizon to double – had compressed from approximately seven months across the 2019-2025 trend to approximately four months in the post-2023 data, with the most recent jump suggesting an even steeper acceleration.⁶ METR itself noted that the confidence intervals are wide – the 95% range for Opus 4.6 spans 6 to 98 hours – and that the benchmark is approaching saturation. But even the conservative end of that interval represents a capability that did not exist twelve months ago.

And then something instructive happened. The discourse, which had moved from feeling to measurement, immediately moved back to feeling. The numbers were so dramatic that the response was not analysis but qualification: the benchmark is saturating, the confidence intervals are wide, scaffold issues affected some tests, these results might not generalise. All of which may be true. But the impulse to qualify – to reach for reasons the numbers might not mean what they appear to mean – is itself a data point. It is what it looks like when measurement outpaces the frameworks people use to interpret measurement.

This oscillation – feeling outrunning measurement, measurement outrunning comprehension, comprehension reaching for caveats to slow the feeling back down – is the lived experience of exponential change. It is what it actually feels like to be inside the curve.

III. The Instruments Are Failing

METR's benchmark is, by the organisation's own admission, beginning to saturate. It is running out of tasks hard enough to differentiate between the most capable models. This is not a failure of the benchmark. It is a success so rapid that the measuring instrument has been outpaced by the thing it measures.

This pattern is not confined to benchmarks. It is everywhere.

Economic models of AI impact – the McKinsey projections, the Goldman Sachs estimates, the OECD scenarios – were calibrated on assumptions about the pace of capability growth that are already obsolete.⁷ Workforce planning tools assume a rate of skill displacement that the METR data suggests is dramatically too slow. Regulatory frameworks – the EU AI Act, the FDA/EMA guidance on AI in drug development – were designed for a technology moving at one speed and are being implemented as it moves at another.⁸ Organisational change methodologies assume that transformation takes quarters or years; the CCRouter story I opened with describes a transformation that took hours.

The instruments we use to understand the world were calibrated for a different rate of change. This is not a criticism of the people who built them. It is a description of what happens when the thing being measured enters a new regime.

The most dangerous consequence is not that the instruments give wrong answers. It is that people continue to trust them because the instruments are familiar and the reality is not. A workforce planning model that says “AI will affect 30% of tasks in your organisation over the next five years” feels measured and responsible. The reality – that AI capability is doubling every few months and that no one knows where the ceiling is – feels hysterical. People choose the measured-feeling answer not because the evidence supports it but because the alternative is vertigo. And vertigo is unpleasant.

But vertigo is also information. The discomfort of exponential change is not a bug in human cognition. It is the correct response to a situation in which the ground is moving faster than your ability to map it. The question is not how to eliminate the vertigo. It is how to act well despite it.

IV. The Level Shift

Return to the six sessions for a moment. The interesting thing about the CCRouter experience was not that AI wrote code faster than I could. That has been true for some time. The interesting thing was that the *nature of my work changed*.

I did not write integration code. I did not run tests. I did not debug failures. I did not even write the tool that enabled the agents to coordinate – Claude Code did most of that work too. What I did was: I defined the problem. I designed the coordination architecture. I decided which agent should have which role. I watched the system operate and intervened when the approach was wrong, not when the code was wrong.

The work moved up a level of abstraction. From writing code to orchestrating agents that write code. From debugging applications to designing systems in which agents debug applications. From solving problems to defining problems clearly enough that agents can solve them.

This is not the first time this has happened in the history of technology. The programmer who moved from assembly language to high-level languages experienced a similar shift. The manager who stopped doing the work and started organising others to do it experienced another. But those transitions played out over years or decades. This one played out over an afternoon.

And here is the part that matters for trajectory: the level I moved to is not a permanent resting place. If AI can be orchestrated to solve integration problems, it can – in principle, and probably soon in practice – orchestrate itself. The supervisory session in my CCRouter setup was a Claude session. There is no technical reason it could not have been configured to design its own coordination strategy, assign its own roles, and manage its own work-

flow without my involvement. I was in the loop because the tools are new and I wanted to understand the process. Not because the process required me.

This is the pattern that makes exponential capability different from linear improvement. Linear improvement makes you faster at the same work. Exponential capability makes the work itself obsolete, then makes the next level of work obsolete, then the next. Each level shift feels like a stable new equilibrium – “AI does the coding, humans do the orchestrating” – until it doesn’t.

The METR benchmark measures autonomous work horizon: how long an AI can work without human intervention. The trajectory – 50 minutes in early 2024, nearly 5 hours by late 2025, 14.5 hours by February 2026 – is a direct measurement of the level shift in progress. Each extension of the autonomous horizon is a reduction in the human role. Not in the value of the human, but in the *necessity* of the human at that particular level of the work.

V. The Asymmetry

In the wake of the METR results, the AI commentator Nathaniel Whittemore made an observation that deserves to be treated as a structural principle rather than a rhetorical point: the cost of overestimating AI vastly exceeds the cost of underestimating it.⁹

If you overestimate AI and prepare accordingly – invest in understanding the tools, restructure your workflows, develop new skills, build organisational capacity for rapid adaptation – and it turns out that AI plateaus or progresses more slowly than expected, you have lost some time and some investment. The skills you developed are not wasted. The organisational capacity is not harmful. You are, at worst, slightly ahead of schedule.

If you underestimate AI and do not prepare – if you assume the current pace will slow, that your profession is somehow exempt, that the benchmarks

don't apply to your domain, that you have years rather than months – and it turns out that the trajectory continues, you face a problem that cannot be solved by catching up. Because the thing about exponential curves is that the distance between you and the frontier does not stay constant while you are standing still. It grows. Every month of inaction is not one month behind. It is one month behind a frontier that has moved further than it moved the previous month.

The seventeenth-century philosopher Blaise Pascal argued that the rational response to uncertainty about God's existence is to live as though God exists – because the cost of being wrong in that direction (a life of unnecessary piety) is trivial compared to the cost of being wrong in the other (eternal damnation). One need not find Pascal's theology compelling to recognise the structure of the argument. It applies whenever the consequences of error are radically asymmetric. This is that situation. You do not need to be certain about AI's trajectory. You need to be honest about what happens if you are wrong in each direction.

The Acceleration Gap – the widening distance between those who are adapting and those who are not – is no longer a strategic concern for the long-term planning horizon. The METR data suggests capability is doubling every few months – and accelerating – which means the gap compounds at a pace that organisational planning cycles cannot match. By the time most organisations have completed a quarterly review of their AI strategy, the capability frontier has moved further than it moved in the entire previous quarter.

VI. The Coding-to-Everything Pipeline

Both episodes of the discourse – Shumer’s essay and the METR results – orbit the same unresolved question: how fast and how completely does AI capability in code translate to capability in everything else?

Shumer’s argument is that code was first because it bootstraps everything else. If AI can write software, it can build the tools that extend its own capability into other domains. The CCRouter story is a small illustration of exactly this: AI wrote the coordination tool that enabled AI to do more work. The capability is self-amplifying.

Saul’s counterargument is that code is uniquely suited to AI – structured, pattern-rich, with clear success criteria – and that messier domains resist automation in ways that benchmarks do not capture. There is truth in this. Writing a legal brief that passes muster in court is different from writing code that passes a test suite. Managing a patient relationship is different from managing an API integration.

But the counterargument understates two things. First, the speed of capability transfer. The same models that write code now also draft legal analyses, generate clinical trial protocols, produce financial models, and write policy recommendations – not at expert level in every case, but at a level that was unthinkable two years ago and that is improving on a timeline measured in weeks. Second, and more importantly, the bootstrapping effect. Every improvement in AI’s ability to write code is also an improvement in AI’s ability to build the tools that extend its capability into non-code domains. The pipeline from code to everything else is not a metaphor. It is an engineering reality.

For the life science industry I work in – the implication is that the timeline for AI impact is not set by the general pace of AI progress. It is set

by the pace at which AI capability in software translates into capability in their specific domain. And that translation is happening faster than most industry planners assume, because the translation itself is being accelerated by AI.

VII. The Transition

The optimists and the pessimists in this discourse have more in common than they realise. The optimists – the heirs of Bastiat, who point to the knitting machine and the power loom and note that new technologies have always created more jobs than they destroyed – have history on their side. The pessimists – those who look at the METR trajectory and Citrini Research’s scenario paper “The 2028 Global Intelligence Crisis,” which triggered an actual stock market selloff when it was published in February – have the unprecedented speed of the current transformation on theirs.¹⁰

Both are probably right about the destination. New industries will emerge. New forms of work will be invented. Human creativity and agency will find new expressions. And also: the transition will be brutal for many people, the benefits will accrue disproportionately to capital owners, and the consumption base that sustains the economy will erode if wages continue to shrink as a share of economic output.

What neither side adequately addresses is the transition itself. Not the before, not the after, but the passage between them. The “liminal period of carnage,” as Whittemore called it.⁹

I have written elsewhere about what happens to identity when the productive capabilities on which professional self-worth is built are no longer uniquely human.¹¹ I have written about the material architecture that is needed when the link between labour and income weakens.¹ But those essays address the philosophical and structural dimensions. The experiential di-

mension – what it actually feels like to live through the transition – is different.

It feels like vertigo. It feels like watching six AI sessions do in hours what would have taken you days, and not knowing whether to be exhilarated or frightened, and settling on both. It feels like reading a benchmark result that says the autonomous work horizon tripled in two months and immediately reaching for reasons it might not be true – not because you have evidence it isn't, but because the alternative is a world that changes faster than you can plan for. It feels like recognising that your skills, your workflows, your professional identity were calibrated for a rate of change that no longer exists.

The vertigo is not the problem. The vertigo is the correct response. The problem is what you do next.

VIII. What the Trajectory Says

The METR data, taken at face value, implies that by mid-2026, autonomous AI work horizons will exceed a standard working day. By late 2026, they may exceed a working week. The confidence intervals are wide and the benchmark is saturating, so the precise numbers are less reliable than the direction. But the direction is unambiguous: the autonomous capability of AI systems on real-world engineering tasks is growing faster than any previous technology capability in recorded history.

This does not mean that AI will replace all human work by 2027. Benchmarks measure narrow capabilities in controlled conditions. Real work is messy, contextual, political, and relational in ways that benchmarks cannot capture. The gap between “can work autonomously for 14 hours on a software task” and “can replace a software engineer” is real and significant.

But the trajectory tells us something important about the *rate* at which that gap is closing. And it tells us that the people, organisations, and institu-

tions that assume they have years to adapt are making an assumption that the data does not support.

I built CCRouter in two hours. Not because I am unusually skilled, but because the tools have reached a point where a coordination service with an embedded MCP server, process-tree identification, message routing, and terminal injection is a two-hour project. A year ago, it would have been a two-week project. Two years before that, it would not have been possible to build at all.

CCRouter is open source.¹² Anyone can run it. Anyone can set up seven terminal windows and watch six Claude sessions coordinate on a shared problem. I mention this not as an advertisement but as an invitation. The experience of watching agents collaborate – of seeing the level shift happen in real time, of feeling the vertigo – is more informative than any benchmark or any essay, including this one.

The curve is not something that happens to other people in other industries at some future date. It is something you can sit inside today, on your own machine, and feel.

The question – the only question that matters now – is what you do with that feeling.

References

1. Stromquist, J. “On Sustenance.” March 2026. Velocity Foundry. Available at [velocityfoundry.org.](https://velocityfoundry.org/) ↵
2. Shumer, M. “Something Big Is Happening.” February 10, 2026. Published at shumer.dev. Reached approximately 82 million views on X. ↵

3. Saul, I. Response to “Something Big Is Happening.” February 2026. [←](#)
4. Wilmanitas. “Tool-Shaped Objects.” February 2026. [←](#)
5. Boyack, C. “AI Isn’t Coming for Your Future, Fear Is.” February 2026. [←](#)
6. METR (Model Evaluation & Threat Research). “Time Horizons” benchmark, updated February 20, 2026. Results: Opus 4.5 (Dec 2025): 4h 49m (50% time horizon); Opus 4.6 (Feb 2026): ~14h 30m (95% CI: 6h to 98h). Overall doubling rate compressed from ~7 months (2019-2025 trend) to ~4 months (post-2023 trend). Available at metr.org/time-horizons/. [←←←](#)
7. See McKinsey Global Institute, “The economic potential of generative AI,” June 2023 (estimated 60-70% of work activities could be automated); Goldman Sachs, “The Potentially Large Effects of Artificial Intelligence on Economic Growth,” March 2023. Both assumed capability trajectories that the METR data suggests were significantly conservative. [←](#)
8. On regulatory frameworks and AI pace, see Stromquist, J. “10 Principles, Zero Excuses: What FDA/EMA’s AI Guidance Actually Means.” January 2026. Velocity Foundry. [←](#)
9. Whittemore, N. “The Perils of the AI Exponential.” The AI Daily Brief, February 24, 2026. [←←](#)
10. Citrini Research (van Geelen, J. and Shah, A.). “The 2028 Global Intelligence Crisis.” February 22, 2026. Framed as a fictional macro scenario memo from June 2028 modelling mass white-collar displacement, “Ghost GDP,” and market collapse. Triggered a significant stock selloff on February 23-24, with IBM experiencing its largest single-day drop in 25 years. Available at citriniresearch.com. [←](#)
11. Stromquist, J. “What Remains.” February 2026. Velocity Foundry. Available at velocityfoundry.org. [←](#)
12. CCRouter is available at github.com/johanstromquist/CCRouter. [←](#)