

Autonomy Without an Invoice: The New Agents' Hidden Ledger

Weekly Analysis — <https://ainews.social>

A tool that suggests is still yours to overrule. A tool that acts has already spent something — a few seconds of a datacenter's electricity, a sip of cooling water, a slice of metered compute — before you get the chance to say no. That is the quiet threshold the industry crossed this season, and it did so without ceremony. The new generation of agents does not wait at the edge of your document offering a phrase. It opens the terminal, edits the file, runs the browser, reviews the code, and reports back that the work is done. The marketing word for this is *autonomy*. The word the marketing omits is *invoice*.

Watch the move in the vendors' own documentation, which is the most honest place to look because it is written for people who have already bought in. Microsoft now ships a system that will read your pull request and post its review directly into the workflow, no longer whispering a suggestion but rendering a verdict [9]. The same capability is rolling into Azure Repos as a preview feature that inserts itself into the moment of merge, the decision point where code becomes product [4]. And Copilot Studio has turned the building of these agents into a publishing pipeline: you design an agent, you publish it, you deploy it across channels, and it runs [3]. Read those three pages back to back and notice what is present — capability, deployment, scale — and what is absent. There is no line item for what any of it costs.

This is not an oversight. It is a genre. The vendor release note, the adoption guide, the productivity module, the code-assist overview: each is a form of writing with its own conventions, and the governing convention is that costs do not appear in the narration until they have become the customer's problem. Kate Crawford named the underlying maneuver precisely — she called it *enchanted determinism*, the framing that invites us "to focus on the innovative nature of the method rather than on what is primary: the purpose of the thing itself," and which "obscures power and closes off informed public discussion, critical scrutiny, or outright rejection" [22]. The agents are enchanting on purpose. The enchantment has a ledger, and the ledger is hidden. This essay is an attempt to read the hidden ledger by reading, closely, what the tools say about themselves and — more revealingly — what they decline to say.

[9] Get started with Copilot code review for pull requests

[4] Copilot Code Reviews for Azure Repos (public preview)

[3] Conceptos clave: Publicar e implementar su agente - Microsoft Copilot
...

[22] The Atlas of AI - Power, Politics, and the Planetary Costs

The Release Note as a Literary Form

Begin with the release note, because it is where a tool tells you who it has become. OpenAI maintains a running document of model release notes, a chronicle of new capabilities, improved reasoning, expanded context windows, and refined behavior [17]. Read it as you would read any account of a thing's arrival in the world. It tells you what the model can now do. It does not tell you what the model now requires: how much more electricity a larger context window burns per query, how much water the cooling loop consumes to keep the inference hardware from throttling, how many accelerators had to be manufactured and shipped and installed to serve the "enhanced" tier. Those are the primary facts of the object's existence, and they are precisely the facts the genre is built to exclude.

This is what it means to say the silence is structural rather than accidental. Crawford's whole argument in *The Atlas of AI* is that these omissions are not gaps in an otherwise complete story; they are the load-bearing walls of the story. The illusions of frictionless, weightless intelligence "distract from the far more relevant questions: Whom do these systems serve? What are the political economies of their construction? And what are the wider planetary consequences?" [22]. A release note that boasted new capabilities *and* disclosed their marginal energy and water draw would be a different kind of document — one that invited the reader to weigh cost against benefit. The industry has chosen not to write that document. When a productivity suite refuses to bolt on generative features at all, it demonstrates that the other choice was always available; the omission elsewhere is therefore a decision, not a limitation of the form.

You can see the same erasure in the humblest corner of the documentation, the part that governs images. OpenAI's collection on image generation walks users through prompts, styles, and outputs [12], and Microsoft's guidance on running DALL·E through its cloud is careful to address the questions that create legal exposure for the buyer — privacy and copyright of generated images [11]. Notice the selection principle. The cost that could land on Microsoft as liability gets a documentation page. The cost that lands on the grid and the watershed gets nothing. The ledger is not blank; it is edited. The vendor discloses exactly the costs that would otherwise become *its* problem and stays silent on the costs it can externalize onto users, institutions, and the planet.

The disciplined reader learns to invert the document. When a tool tells you what it can do, ask what it now consumes to do it. When a release note announces enhancement, ask enhancement measured

[17] Notas de lanzamiento de modelos
- OpenAI Help Center

[22] The Atlas of AI - Power, Politics,
and the Planetary Costs

[12] Image Generation - OpenAI Help
Center

[11] How to ensure privacy and copy-
rights for images generated via Dall-e

against what, at what marginal draw, disclosed where. The absence of an answer is itself the finding. A capability claim with no cost claim attached is not a neutral description; it is a rhetorical structure designed to make the cost unthinkable at the moment of adoption.

The Productivity Promise, Unaudited

If autonomy is the headline, productivity is the pitch. This is the dominant framing of the entire tools discourse — by the measure of this week’s corpus, roughly a quarter of the coverage treats AI as a utility, a device for getting more done, and the vendors supply the vocabulary. Microsoft offers a training module whose title is the thesis of the whole category: boost your productivity with Copilot [2]. Its French counterpart promises to *unlock* productivity with generative AI, as though productivity were a resource sitting behind a door that only the tool can open [7]. The claim is everywhere, and it is stated with a confidence that outruns the evidence offered to support it.

Here the skeptical adopter should slow down and ask the evaluator’s question: what is the measurement? The vendors have an elaborate apparatus for tracking their tools, but read what it actually measures. Microsoft’s Copilot adoption report is built to show organizations how many seats are active, how frequently the tool is invoked, how deployment is progressing [15]. The onboarding guide for IT administrators is similarly framed around driving usage — enabling, encouraging, and sustaining engagement across the organization [14]. This is a measurement regime, and it is precise, but it measures *adoption*, not *value*. It counts how many people used the tool, not whether the work got better, faster, or cheaper in any way that survives an honest accounting of the time spent prompting, checking, and correcting. A metric of usage is not a metric of benefit. Conflating the two is the oldest trick in the productivity literature, and the documentation performs it fluently.

Meredith Broussard’s *Artificial Unintelligence* offers the corrective anecdote, and it is worth holding onto because it captures the gap between the fancy claim and the delivered result. She describes a choice her mother faced between an inexpensive option “that used the fancier technology that didn’t do the job she wanted” and a more expensive option “that used the less-fancy technology that did exactly the job she wanted” [22]. The productivity pitch assumes the fancy technology does the job. The careful adopter knows that the fanciness of a method and its fitness for a purpose are independent variables, and that the burden of proof lies on the party making the claim. That

[2] Aumentar la productividad con Microsoft Copilot

[7] Déverrouiller la productivité avec l’IA générative - Training

[15] Microsoft Copilot adoption report | Microsoft Learn

[14] Microsoft Copilot adoption and onboarding guide for IT admins

[22] Artificial Unintelligence - How Computers Misunderstand

burden is not discharged by an adoption dashboard.

Broussard also supplies the reality check the vendors leave out, from her own experience inside real systems: her small mistakes, because "precision matters in computation," were "probably creating a cascade of other inexplicable, frustrating errors for other people" [22]. Transpose that into the world of the acting agent. A tool that suggests a bad line of code costs you the moment it takes to reject it. A tool that edits the file, runs the terminal, and merges the review — the tool Microsoft is now shipping into the pull-request workflow [9] — commits its errors into shared systems where they cascade. The productivity gained at the point of action can be lost several times over at the points of downstream failure, and none of that appears in the seat-count. The honest question a buyer should demand of any productivity claim is not "how many people used it" but "what did the errors cost, and who absorbed them." The documentation is not built to answer.

[22] Artificial Unintelligence - How Computers Misunderstand

[9] Get started with Copilot code review for pull requests

Deprecation Is the Business Model

The clearest window onto the hidden ledger is not a marketing page at all; it is a support forum, where the costs the vendor externalized surface as the customer's confusion. Consider the developers who logged into their tooling one day to find that the models they had built on were gone. "GPT-4o and GPT-4.1 were deprecated today," reads one question thread, the voice of someone whose working system had just been altered by a decision made elsewhere [10]. Another asks the anxious forward-looking version of the same question: is GPT-4.1 going to be removed from the API entirely, and what happens to everything built on top of it [13]? These are not complaints about capability. The new models are, by the vendors' account, more capable. These are the sounds of dependence discovering its own terms.

[10] GPT 4o and GPT 4.1 were deprecated today - Microsoft Q&A

[13] is GPT 4.1 gonna be removed in API usage? - Microsoft Q&A

This is where the political economy of autonomy becomes legible. The pitch is that the agent frees you; the structure is that the agent binds you to the platform's release cadence. When the vendor deprecates a model, the migration cost — rewriting prompts, revalidating outputs, retesting the whole pipeline — lands entirely on the customer, and it lands on a schedule the customer does not set. The tool that was sold as radical independence turns out to require the user to "deepen their dependence on the vendor's next release," to borrow the frame of the wager. The autonomy is real, but it belongs to the platform, not to you. You get the automation; the platform keeps the control.

Google’s version of the same dynamic is metering. The support page governing upgrades and limits for Gemini subscribers is, functionally, the document that tells you where your autonomy stops — how many high-end queries you may run before the tool throttles you back to a lesser tier [16]. Read it alongside Google’s announcement of expanded AI access across Workspace [1] and the shape of the arrangement resolves: access is expanded and access is capped, and the vendor holds both dials. The “significantly enhanced capabilities” you were sold arrive inside an envelope the vendor can shrink at will. The invoice you never saw at purchase is being paid in installments of metered constraint.

There is a genuinely revealing tell in how the platforms describe change itself. Microsoft maintains a “what’s new” page for Copilot Studio, a rolling feed of additions and revisions to the agent-building platform [18]. The genre of the changelog presents perpetual motion as pure gain — every entry is an improvement, an addition, a new power granted. What the changelog never frames as a cost is the labor it imposes on everyone downstream who must now learn, migrate, and revalidate. The vendor’s forward motion is the customer’s treadmill. And the customer runs on it not because each release is worth the disruption but because stepping off means abandoning the platform entirely — the dependence, once established, converts every “improvement” into an obligation. Crawford’s point about closing off “outright rejection” as an option [22] is not abstract here; it is the lived condition of anyone whose stack rests on a model that can be deprecated by press release.

The Costs That Land on the Customer

Follow the money and the pattern holds across every level of the stack. At the enterprise scale, the promise is that agents will absorb work and therefore cost. Microsoft’s business FAQ for Copilot frames the tool in exactly these terms — capability delivered, work offloaded, value returned [20]. But the enterprise reality this season has been the discovery that autonomy has a running meter, and that the meter runs faster than budgeted. When an organization exhausts its entire annual AI allocation months before the year is out — the pattern reported of firms whose spend outran their forecasts by spring — the lesson is not that the tools failed to work. It is that the cost of the tools working was never on the table when the decision to adopt was made. The invoice arrived after the autonomy did.

The deployment documentation makes this asymmetry concrete.

[16] Mises 00e0 niveau et limites des applications Gemini pour les abonn00e9s ...

[1] AI Expanded Access - Google Workspace Learning Center

[18] Novedades de Copilot Studio - Microsoft Copilot Studio

[22] The Atlas of AI - Power, Politics, and the Planetary Costs

[20] Preguntas m00e1s frecuentes sobre la empresa Microsoft Copilot

Rolling Copilot out to an organization is presented as a project the customer manages — minimum requirements to meet, prerequisites to satisfy, a rollout to sequence [21]. Deploying the app is likewise the customer’s undertaking, with the customer’s IT staff absorbing the integration labor [6]. Every one of these steps is real work, and every hour of it is a cost that the vendor’s productivity pitch quietly assigns to the buyer while crediting the tool with the savings. The accounting is one-sided by construction: the benefits accrue to the tool’s story, the costs accrue to the customer’s operations.

This is where the specific counterfactuals of the season do their argumentative work. When an enterprise deploys twenty-five supply-chain agents, the headline is capability at scale; the ledger is the compute, the oversight, and the failure-recovery those agents demand, all borne by the enterprise. When a productivity suite pointedly refuses to embed generative AI, it is not falling behind — it is declining to externalize a cost onto its users, and in declining it proves the externalization elsewhere was a choice. Broussard’s framing of her mother’s dilemma is the template: the fancy option that does not do the job versus the plain option that does [22]. The suite that refuses AI is choosing the plain option on the user’s behalf. The agent platforms are choosing the fancy option on their own.

And there is a deeper cost the tool framing cannot register at all, which is mission. When a company founded to develop artificial intelligence “for the benefit of humanity” is credibly accused of abandoning that founding purpose, and the coverage metabolizes the charge down to a procedural squabble over a delayed proceeding, something has been erased that no adoption dashboard could ever capture. The substantive question — whether the institution still serves the public good it invoked to justify its existence — gets converted into a scheduling matter. This is the same editing principle at work in the release note, applied to political economy rather than to energy: the cost that would indict the vendor’s architecture is reframed as someone else’s smaller, more manageable problem. Crawford’s questions — whom do these systems serve, what are the political economies of their construction [22] — are exactly the questions the narrative is organized to prevent you from asking.

Reading the Tool for What It Won’t Say

So what should a careful adopter actually do, standing in front of an agent that promises to act on their behalf? The first discipline is to read the tool’s self-description as a document with a point of view

[21] Rollout Microsoft Copilot to your organization

[6] Deploy the Microsoft Copilot App | Microsoft Learn

[22] Artificial Unintelligence - How Computers Misunderstand

[22] The Atlas of AI - Power, Politics, and the Planetary Costs

rather than as a specification sheet. When Gemini Code Assist describes itself as an overview of capabilities and integrations [8], or when its enterprise codelab walks through customizing code generation to a company's own repositories [19], the text is answering the question "what can this do for me?" It is not answering "what does this do to me, and to the systems and people around me?" The second question is the one that matters for evaluation, and the reader has to supply it, because the genre never will.

The second discipline is to distinguish the two stances that structure the entire coverage. There is the PRO_AI voice — fluent, confident, native to the release note and the productivity module, treating each new capability as self-evidently good [2]. And there is the skeptical voice, which in this corpus rarely gets to organize a document of its own and instead leaks through in the support threads, where the deprecation lands [10], and in the limits pages, where the metering bites [16]. The imbalance is itself information. When the promotional voice owns the overview pages and the skeptical voice is confined to the troubleshooting forums, the discourse is telling you where the vendors want your attention and where they do not. A reader who only consults the overviews will never meet the costs; a reader who reads the forums first will meet nothing else.

The third discipline is to insist on the material floor beneath the abstraction. Every claim of frictionless capability rests on hardware, power, and water, and the fact that a Copilot for Power BI can generate a report from a prompt [5] does not suspend the physics of the datacenter that served the request. Crawford's insistence on the "planetary costs" is not a moral flourish appended to the technical story; it is the technical story, the part the enchantment is designed to make invisible [22]. The adopter who keeps the material floor in view is harder to enchant, and being harder to enchant is the whole of what critical evaluation amounts to here.

What questions, then, should travel with every agent you are asked to adopt? Ask what it consumes per action, and where that figure is disclosed. Ask who absorbs the cost when it errs — because it will err, and errors from an acting agent propagate further than errors from a suggesting one [9]. Ask what happens to your work when the underlying model is deprecated [13], and who sets the schedule. Ask whether the productivity claim was measured as value delivered or merely as seats activated [15]. None of these questions is answered by the marketing, and each of them can be answered — the counterfactual of the tool that refuses AI proves that a company can choose to answer them. The silence is not imposed by the medium. It is a product decision.

[8] Gemini Code Assist overview | Google for Developers

[19] Personalizaci00f3n de c00f3digo con Gemini Code Assist Enterprise

[2] Aumentar la productividad con Microsoft Copilot

[10] GPT 4o and GPT 4.1 were deprecated today - Microsoft Q&A

[16] Mises 00e0 niveau et limites des applications Gemini pour les abonn00e9s ...

[5] Copilot for Power BI overview

[22] The Atlas of AI - Power, Politics, and the Planetary Costs

[9] Get started with Copilot code review for pull requests

[13] is GPT 4.1 gonna be removed in API usage? - Microsoft Q&A

[15] Microsoft Copilot adoption report | Microsoft Learn

The Silence Is the Product

Here is the synthesis the week's evidence supports. The new agents are genuinely more capable than what came before; the code review that posts its verdict into the pull request is real [4], the agent you can publish and deploy across channels is real [3], the expanded access is real [1]. This is not an argument that the tools do nothing. It is an argument that what they do has a cost, and that the systematic non-disclosure of that cost is the most consequential design choice the vendors have made — more consequential, in the long run, than any capability on the release note.

The cost does not vanish when it goes unmentioned. It relocates. It moves from the vendor's balance sheet to the grid and the watershed, where it becomes the planet's problem in a form no user ever consents to. It moves from the release note to the support forum, where it becomes the developer's problem the morning a model is deprecated [10]. It moves from the productivity module to the IT department, where it becomes the rollout the customer must staff and sequence [21]. And it moves from the annual forecast to the crisis meeting when the budget is exhausted by spring. At every relocation the same thing happens: control stays with the platform, cost flows to everyone else, and the narration credits the tool with pure benefit. This is not a bug in the story. It is the story's purpose.

Broussard's small confession — that her own errors, because precision matters, cascaded into "inexplicable, frustrating errors for other people" [22] — is the whole essay in miniature, scaled up to the level of an industry. When the acting agent errs, the errors cascade, and the people who inherit them rarely know where they came from, because the tool that produced them was described to them as autonomous, capable, and free. Crawford's warning that enchanted determinism "closes off informed public discussion, critical scrutiny, or outright rejection" [22] is the diagnosis of why we let this happen: we cannot scrutinize a cost we have been trained not to see, and we cannot reject a tool whose price has been kept off the invoice until after we depend on it.

The remedy is not technical and it is not even, in the first instance, regulatory. It is a reading practice. Treat the release note as a document that edits [17]. Treat the productivity claim as unproven until measured against value rather than usage [15]. Treat the deprecation notice as the true statement of who holds control [13]. Treat the metering page as the real map of your autonomy's edges [16]. Above all, when a tool offers you autonomy without an invoice, understand that the invoice exists, that someone is paying it, and that the vendor has

[4] Copilot Code Reviews for Azure Repos (public preview)

[3] Conceptos clave: Publicar e implementar su agente - Microsoft Copilot ...

[1] AI Expanded Access - Google Workspace Learning Center

[10] GPT 4o and GPT 4.1 were deprecated today - Microsoft Q&A

[21] Rollout Microsoft Copilot to your organization

[22] Artificial Unintelligence - How Computers Misunderstand

[22] The Atlas of AI - Power, Politics, and the Planetary Costs

[17] Notas de lanzamiento de modelos - OpenAI Help Center

[15] Microsoft Copilot adoption report | Microsoft Learn

[13] is GPT 4.1 gonna be removed in API usage? - Microsoft Q&A

[16] Mises 00e0 niveau et limites des applications Gemini pour les abonn00e9s ...

decided you should not look. The tools are choosing their silences. The silences are the real product. And the one power the reader retains — the power the enchantment is built to dissolve — is the power to keep reading the ledger the tool would rather you never open.

References

1. AI Expanded Access - Google Workspace Learning Center
2. Aumentar la productividad con Microsoft Copilot
3. Conceptos clave: Publicar e implementar su agente - Microsoft Copilot ...
4. Copilot Code Reviews for Azure Repos (public preview)
5. Copilot for Power BI overview
6. Deploy the Microsoft Copilot App | Microsoft Learn
7. Déverrouiller la productivité avec l'IA générative - Training
8. Gemini Code Assist overview | Google for Developers
9. Get started with Copilot code review for pull requests
10. GPT 4o and GPT 4.1 were deprecated today - Microsoft Q&A
11. How to ensure privacy and copyrights for images generated via Dall-e
12. Image Generation - OpenAI Help Center
13. is GPT 4.1 gonna be removed in API usage? - Microsoft Q&A
14. Microsoft Copilot adoption and onboarding guide for IT admins
15. Microsoft Copilot adoption report | Microsoft Learn
16. Mises et limites des applications Gemini pour les abonnés ...
17. Notas de lanzamiento de modelos - OpenAI Help Center
18. Novedades de Copilot Studio - Microsoft Copilot Studio
19. Personalización de código con Gemini Code Assist Enterprise
20. Preguntas frecuentes sobre la empresa Microsoft Copilot
21. Rollout Microsoft Copilot to your organization
22. The Atlas of AI - Power, Politics, and the Planetary Costs