

BEYOND ACCESS

Access Is Not Adoption

Why the gap between the people getting real leverage from AI and the people merely paying for it has almost nothing to do with which tool they chose

Giovanni Leonardi

January 2025



Access Is Not Adoption

Transformation · No. 336

Originally written January 2025 | Re-edited and re-rendered for the WhereBeyond Edition,
2025

*"It moved their attention off the work a machine could
carry and onto the work only they could."*

1. The desk with everything open

Picture a capable person at the start of this year. They have paid, personally, for the two leading general-purpose assistants. Their employer has switched on the one baked into the office suite. They have a note somewhere with a dozen clever prompts a colleague swore by, and a browser tab open to a newsletter promising the ten that will change everything. By any measure of access, they are fully equipped.

Now look at their working week. It is, in almost every respect, the week they had a year ago. The same meetings, the same drafting, the same slow triage of an overfull inbox. The assistants get used — a paragraph rewritten here, a summary there, the occasional flash of something genuinely useful — but nothing structural has changed about how the work gets done. The tools are being consumed, not operated.

I have watched this pattern repeat across enough desks now to be confident it is not about the person's ability or their appetite. It shows up most in the sharpest people, the ones who accumulate the cleverest prompts fastest — because a clever prompt is a trophy you can collect without changing anything, and the folder of them grows while the shape of the week stays flat. What such a person is missing is not a tool and not a prompt. It is an operating model: a deliberate, personal answer to the question of what they now delegate, how they check it, and how their own working method has to change to absorb the help.

2. The seduction of the tooling question

The reason the operating model goes unbuilt is that a much easier question stands in front of it, wearing the same clothes. Which tool should I use? is tractable. It has an answer you can buy. You can compare the assistants, read the comparisons other people have written, pick one, and feel that you have engaged with the shift. The purchase produces the sensation of adoption without any of its substance.

Building an operating model is the opposite kind of problem. It cannot be bought, it resists being copied from someone else, and it delivers nothing on the day you start. It asks you to look honestly at how you actually spend your cognitive hours, decide which of them are genuinely yours to keep and which are drafting and first-pass work you have

simply always done by hand, and then rebuild your habits around that split. That is uncomfortable and slow, and it competes for exactly the reflective time the overfull week never surrenders. So the tooling question wins, again and again, because it is the one that can be closed.

Access is now close to universal and roughly equal. Two people on the same team hold the same licences and, underneath them, the same capability. The entire difference in what they get out of it is the operating model each has or has not built. That is where the leverage lives, and it is the one part no subscription includes.

3. The case for waiting

There is a serious version of the opposite argument, and it deserves to be met at its strongest rather than waved away. It runs like this: the tools are not yet trustworthy. They still fabricate with great confidence. The newer reasoning models are slower and more expensive, and even they are wrong often enough that anything that matters has to be checked by hand — at which point, why not simply do it by hand? The disciplined move, on this view, is to wait: let the capability mature, let the reliability climb, and adopt in earnest when the tools have earned it.

Everything in that argument about today's reliability is true. The conclusion is still wrong, and the reason is a matter of what improves on its own and what does not. Capability improves on its own. It has been improving, visibly, quarter over quarter, with no effort required from the person waiting. What does not improve while you wait is your operating model — the routing instinct, the verification discipline, the reshaped week. Those are muscles, and they are built only by use. When the next jump in capability arrives, and it will, it lands as pure leverage for the person who already has the muscle and as a fresh, unopened box for the person who waited. Waiting does not spare you the learning curve. It just guarantees you climb it late, in a hurry, and behind the people who started while the tools were still visibly imperfect.

“The tools will keep getting better whether or not you are ready. Your readiness is the only variable you actually control.”

4. What the model actually is

An operating model sounds grander than it is. In practice it is three habits, held together.

The first is a routing discipline — a standing, honest sort of your own work into what you keep, what you delegate, and what you delegate but always verify. Consider a concrete

week. A senior manager who tracked one found something close to a hundred and twenty discrete cognitive tasks: drafting, summarising, analysing, deciding, and a great deal of triage. Roughly forty of those were first-pass drafting and summarising — laborious work, but not, in truth, where their judgement earned its keep. Before, all forty were done by hand. After a few weeks of building the habit, around twenty-five were being routed to an assistant with a deliberate check on the way out. The reclaimed time came to something like six hours in the week.

The six hours are not the point, and this is where most productivity framing goes wrong. The point is where those hours went: into the five or six genuinely hard calls of the week that had previously been squeezed into twenty rushed minutes each. The model did not make the person faster at everything. It moved their attention off the work a machine could carry and onto the work only they could.

The second habit is verification calibrated to stakes. The two failure modes are equal and opposite: trust everything, and you ship confident errors with your name on them; trust nothing, and you do all the work twice and gain precisely nothing. Neither is an operating model. The discipline is knowing, task by task, how much checking a given output actually warrants — a throwaway internal note and a figure bound for a board paper do not deserve the same scrutiny — and building the check into the flow rather than bolting it on afterwards.

The third is a feedback loop. What you delegate this month is not what you will delegate next. The boundary moves as the tools change and, more to the point, as you get better at reading their edges — where they are strong, and where they mislead you quietly enough that you would not notice unless you were looking. An operating model that is never revised is not a model; it is a one-time setup already going stale.

5. Governing what you have not operated

For anyone leading others, this stops being a matter of personal productivity and becomes one of governance. A leader who has not built their own operating model is not thereby exempt from the question — they are simply answering it by proxy. They will issue a policy, approve a licence spend, perhaps commission a pilot, and believe they have led the organisation through the shift.

But you cannot govern well what you have never personally operated. The leader who has done the work knows in their own hands where these tools are strong and where they mislead, what a realistic productivity claim sounds like and what is theatre, which of their people are genuinely finding leverage and which are performing enthusiasm. The leader who has not done the work has to take all of that on faith, and will tend to be sold the version that is easiest to present. We are being asked to govern a capability many of us have not yet learned to use ourselves — and that gap, not the technology, is the real exposure.

6. Start with one loop

None of this is an argument for a grand personal transformation programme, which would fail for the same reason the overfull week never yields its reflective hours. It is an argument for one loop. Take a single recurring task you already do by hand and do not enjoy — the weekly summary, the first draft of the standing report, the initial pass through a long document. Route it deliberately. Build the verification step in. Notice, honestly, what came back well and what did not, and adjust. Then, once that loop runs without your thinking about it, widen it to the next task.

The people who will look, a year from now, as though they have some rare aptitude for this will almost never be the ones who found the best tool or hoarded the cleverest prompts. They will be the ones who quietly built the muscle while the rest of us were still comparing subscriptions. Access was never the thing that separated them. The operating model was — and that, unlike the access, was always theirs to build.