The Invisible Tax: Cognitive Labor in Systems

by Abby Buchanan

A lot of people think the hard part of work is having the idea.

Often it isn't.

The hard part is turning the idea into something a system can recognize.

A thought becomes a slide deck.
A pattern becomes a report.
An insight becomes documentation.

A strong instinct has to become a clear explanation—in the right tone, in the right sequence, in a format someone else can quickly scan and approve.

That translation layer takes an enormous amount of energy, and most of the time it goes unnamed.

We tend to talk about work as if the value lives in the visible output: the presentation, the brief, the summary, the ticket. But those artifacts are often not the thinking itself. They're the packaging required to make thinking legible inside a system.

That distinction matters.

Some people naturally translate their thoughts into these formats. Others don't. That doesn't mean they have less insight. It means systems reward a particular kind of fluency.

A lot of what gets mistaken for competence is really compatibility.

If your thinking is linear, structured, and institution-friendly, you're more likely to be read as clear, strategic, and capable. If your thinking is nonlinear, associative, intuitive, or difficult to flatten into bullet points on demand, the same level of intelligence can be much harder for a system to recognize.

This is one reason so much talent gets missed.

People assume the best ideas rise to the top. I don't think that's always true. Often the ideas that surface most easily are simply the ones that translate cleanly into the language a system already knows how to reward.

That creates a kind of invisible tax.

You see it in the extra effort it takes to write something that sounds "professional."
In the energy required to reorganize a rich, messy insight into something polished and defensible. In the constant work of making yourself legible.

For some people, that cost is minor. For others, it's enormous.

And not all thinking moves through formal systems in the same way.

In some environments, ideas move through relationships—through conversation, proximity, trust, familiarity. Some people can say something casually in a room and have it taken seriously without needing to formalize it into a structured artifact.

Others don't have access to those spaces. Or don't move comfortably within them.

In those cases, formal systems become the only pathway to being understood. The idea has to be written down, organized, justified, and shaped into something that can stand on its own.

That difference matters too.

Some people are able to bypass the translation layer entirely, while others are required to move through it every time.

That gap isn't about intelligence. It's about access—and the conditions under which someone is expected to make themselves understood.

This is part of why I keep coming back to AI.

A surprising amount of what AI is good at is not thinking. It's translation.

Structuring text. Summarizing material. Reorganizing ideas. Adjusting tone. Turning rough thought into something more linear and legible.

That doesn’t mean AI is replacing human intelligence. If anything, it may be exposing how much of modern work was never really about intelligence in the first place.

It was about formatting, sequencing, and managing the distance between a living mind and a rigid system.

That’s what makes this moment interesting.

If AI can reduce some of that translation burden, it could allow more people to contribute in ways that were previously blocked by friction.

But it could also reinforce the problem.

If systems simply raise the bar—expecting more polish, more output, more responsiveness—then the invisible tax doesn’t disappear. It recalibrates.

That’s the tension I keep thinking about.

Not whether AI is good or bad.

But whether it changes whose thinking gets to count.

Because most systems are not designed to recognize intelligence in all its forms. They’re designed to recognize familiarity, fluency, and alignment with a particular structure.

Once you notice that, you start seeing it everywhere.

The question isn’t only how we make tools more efficient.

It’s whether we can design systems that ask less translation of people in the first place.