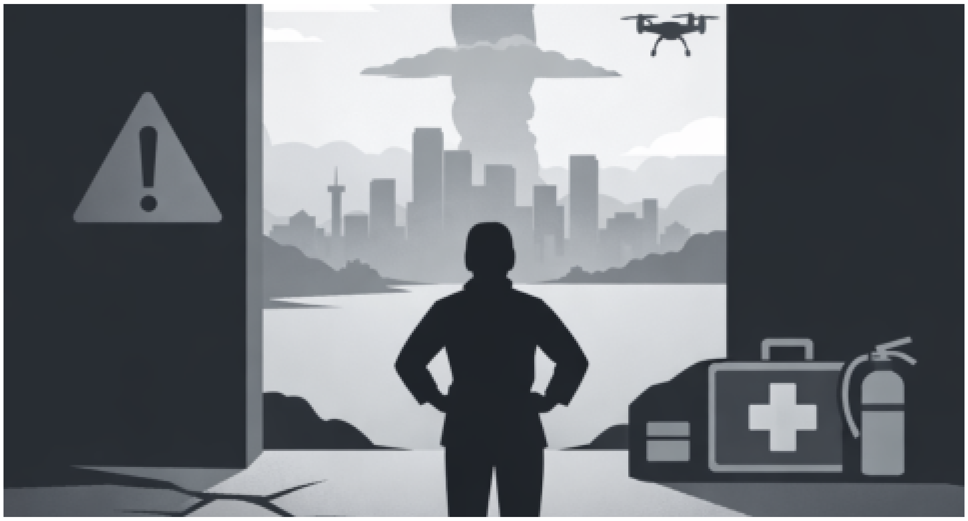
Head of "We Should Probably Not Do That"

Mon, 02/02/2026 - 09:00
|
Blaine Fisher, Ph.D.

View PDF



A job posting tells a truth. It arrives in plain language without fanfare, which is usually how important truths show up when you are actually looking.

OpenAI posted a role called Head of Preparedness.

I keep thinking about the title. It's not glamorous. Nobody puts "Head of Preparedness" on a conference lanyard to impress people. But a company building frontier models chose to fund a leader whose work centers on risk, with real authority and a real budget. That choice says someone there understands what happens when power leaves the lab and lands in the hands of people using tools in ways you did not plan for and cannot fully steer.

Innovation brings risk. Power meets real life, and real life brings users with conflicting incentives, strange edge cases, and adversaries who look for the space between intent and outcome because that is where opportunity lives. No technology gets around this. You can design a perfect engine, but you can't design the driver who decides to see what happens if they shift into reverse at seventy miles an hour. AI fits into that pattern because the pattern is about power moving through human systems.

Preparedness starts when you stop pretending you can build powerful things without someone getting hurt.

It has two jobs. One is obvious: anticipate the harms you can already see coming if you pay attention. The other is harder: respond well when something happens that you didn't predict and can't neatly categorize. Emergency management taught me this early, back when I still believed a good plan could cover everything. Experience corrected that. Every plan I've ever written has been wrong about something important. So you hope for the best, and you prepare for the worst because you actually live in the world. Shrink the blast radius. Practice failing in controlled conditions so the first time people feel pressure is not the first time they're making real decisions.

Frontier AI needs that mindset.

The familiar fear story is a robot uprising: a superintelligent system taking control, a clean sci-fi arc with a twist ending. My worry lives in duller places, where policy meets deployment, literacy meets confusion, and good intentions scale into harm.

I've watched a version of that dynamic show up in education in ways that are almost painfully ordinary.

Last year a faculty member forwarded me an email with a subject line that was basically a sigh: "Is this going to be a problem?" Attached was a student's discussion post. It was good. Too good. Smooth tone, tidy citations, confident structure, the kind of writing that reads like it was sanded and varnished.

The instructor had a hunch. They pasted the prompt into a public model and got back something close, same moves, same arc, same polite conclusion. Not identical, but close enough to make the instructor's stomach drop. Because now what? Accuse a student based on a vibe? Let it slide and teach them nothing? Redesign the entire course mid-semester?

We got on a call. It started practical and turned personal fast.

The instructor didn't want to play detective. They weren't even angry at the student, at least not in the way people imagine. They were angry at the situation. They said, "I feel like I'm either going to become a cop or I'm going to give up. Neither is why I teach."

We worked through options. Not "solutions," because that word is too clean. Options. The instructor rewrote the next assignment so students had to connect the reading to a specific moment from class discussion, with a quote from their own notes. They added a short oral check-in for a subset of students, five minutes each, which is the kind of thing that sounds simple until you try to schedule it. They changed the rubric to reward process and drafts and revision notes. They made it harder to outsource the whole thing without showing your work.

It helped. Mostly.

A month later, another edge case. A student used a model to generate a "reflection" on their learning. It was warm and thoughtful and also, once you knew what to look for, clearly not theirs. When the instructor asked, the student admitted it casually, like it was the same as using spellcheck. They genuinely didn't understand why it mattered.

That moment sits with me more than the "cheating" story, because it's quieter and harder. It's not a villain story. It's a literacy story. It's a norms story. It's a "what are we even doing here" story.

Now scale that dynamic outside academia and you get failures nobody plans for.

Here's one I keep thinking about as I explore AI tools that could eventually plug into university systems. Tulane's tech stack includes platforms like ServiceNow for student support tickets, and I can already see the pitch: an AI summarizer that reads incoming tickets, summarizes the issue, suggests a response, saves staff time. Everyone is under-resourced and tired. The idea would be straightforward. It would probably even work, mostly.

But I can also see the failure mode. The summary becomes the truth.

Picture this: a ticket comes in from a student who writes, in a messy, emotional paragraph, that they can't access a proctored exam portal and they're panicking. The model summary turns it into: "Student requests assistance logging in; issue not urgent." The staff member triaging the queue sees "not urgent," clicks it into the regular bucket, moves on.

The student misses the exam window. They file a complaint. The complaint lands, eventually, in a meeting that was supposed to be about "tool adoption" and turns into an argument about "accountability." Everyone does the predictable thing. The vendor blames configuration. The staff member blames the tool. The tool did what it was trained to do: compress a messy human message into a neat, calm sentence. The student just wants someone to say, plainly, "We messed up and we're sorry."

That's harm that looks small in a report and feels huge when it's your exam, your semester, your money.

And it's what makes people start to say, "I don't trust this AI stuff," even if their complaint isn't really about AI. It's about being dismissed. It's about the system taking the easy interpretation and treating it as fact.

OpenAI hiring a Head of Preparedness is a signal that someone there is thinking about those moments before they happen at scale.

Because once a model isn't just answering questions but touching workflows, the shape of risk changes. The mistakes get slippery. They don't show up as "the model hallucinated." They show up as someone's request being down-ranked, someone's warning being smoothed out, someone's case being routed the wrong way because the summary sounded confident.

There's a reason I keep coming back to meetings, too. The real damage often isn't the initial error. It's what happens in the room afterward.

A few months ago, in a different context, I sat in a meeting where a department wanted a campus-wide AI policy "by Friday." That was the phrase. By Friday.

Because trustees were asking questions and parents were sending emails and someone had read a scary article. The people in the room had wildly different problems. One person needed something to hand to accreditation. Another needed a sentence they could paste into a syllabus. Someone else needed a way to stop their grading workload from exploding. Everyone wanted one clean rule that would make the anxiety go away.

It doesn't exist. The rule doesn't exist.

You can either pretend it exists and write something broad that nobody follows, or you can admit the truth, which is that you're going to need a few rules, plus teaching, plus enforcement, plus exceptions, plus constant revision. People hate that answer. They also recognize it.

That's what preparedness feels like in practice. You are the person in the room saying, "We can't get out of this with one sentence."

If I walked into a Head of Preparedness role, the first weeks would probably feel like that. Less like strategy and more like triage. Lots of people wanting certainty. Lots of people wanting to ship. Lots of people wanting someone else to own the scary parts.

And I'd be lying if I said I wouldn't sometimes be annoyed. There's a particular kind of frustration that comes from watching smart people treat "we'll handle safety later" as a reasonable plan. Later turns into never. Or later turns into "after the incident." That's the worst version of later.

So you start where the mess is.

One launch I observed in a different organization, not OpenAI, involved a last-minute scramble because someone discovered that the model behaved differently in production than it did in testing. In testing, the tool had access to a cleaned dataset. In production, it was reading from a live system with messy fields and weird edge cases. It started making confident recommendations based on incomplete records. A junior analyst flagged it in a thread that got buried under a flood of launch-day chatter. The fix ended up being a manual "hold" on the feature flag. The person who had the authority to do that was, inconveniently, on a flight.

That's what "who can stop this" actually means. Not a theoretical process. A person with the right permissions, in the right time zone, paying attention.

Once you've lived through that once, you stop caring about elegant frameworks. You care about dull things: permissions, escalation paths, which Slack channel gets monitored, whether people know what "stop" even looks like.

That's also where "paper safety" gets exposed. You find out quickly whether your safety documentation is a real tool or a folder of PDFs nobody opens.

In that ServiceNow scenario, the obvious fix wouldn't be philosophical. It would be operational. You'd change the UI so the model summary couldn't overwrite the student's original text on the first screen. You'd add a big "raw message" toggle that's on by default. You'd also add a simple rule in the triage workflow: if a ticket includes certain keywords, the system flags it as high priority regardless of what the summary says. The keywords list would be hilariously imperfect. It would catch "panic" and "exam" and also catch a student joking about "panic buying snacks." Still, it would shift staff behavior back toward reading the human message instead of trusting the compressed version.

That tweak isn't glamorous, but it's the whole game. It's also what a Preparedness leader can institutionalize: "When the model is confident, you don't automatically trust it more. You trust it differently."

From there, the work spreads outward, but not in a neat ladder.

You need a shared understanding of what "frontier" means inside the org, otherwise the term becomes a status badge instead of a trigger for heightened scrutiny. Then you need the basic artifacts—documents that exist for every major launch so you aren't inventing the safety story from scratch every time. You need a place where the major risks are tracked, with owners and current coverage, because "everyone knows" is not a control.

The plumbing matters just as much. Evaluation routines should be stable enough that teams can run them without begging a specialist for help. Monitoring needs to catch weird patterns early enough to matter. And when the bad thing happens, you want incident response that doesn't rely on improvisation. You want the ability to pause or roll back, and you need that button to be real, not ceremonial.

But even saying it that way starts to sound like a checklist, and checklists can become comforting lies. The real test is boring: do people actually use the tools when it's inconvenient? Do they slow down when the results are ugly? Do they accept friction when friction is the price of not hurting someone?

Threat modeling is a good example. People talk about it like it's an academic exercise. It's not. It's you sitting down and admitting, in detail, how someone could abuse what you're building. It's also you admitting where you're guessing.

A student in one of my classes once asked, "Why are you so focused on misuse? Isn't that pessimistic?" And I didn't have a clean answer for them. I said something like, "Because I've seen what happens when you assume people will behave well." That sounded harsher than I meant. I regretted it a little. Then I doubled down anyway, because the point stands.

The better version is probably this: if you're building something powerful, you're building something that will be used by people who don't share your intentions. That's not cynicism. That's just noticing the world.

In practical terms, that means your safeguards have to tie back to specific pathways. Otherwise you end up with a pile of controls that look impressive and don't match real risk. You also have to be careful not to publish the sort of detail that turns your safety work into a user manual for attackers. That line is uncomfortable. You'll get it wrong sometimes. You just try to get it wrong in a way that doesn't make things worse.

At some point launches turn into case files. Not literally, but emotionally. You want to know what changed, what you tested, what you didn't test, what still scares you, what you'll watch after shipping, and what you'll do if you see the wrong pattern. You want dissent written down, too, because the most depressing postmortems are the ones where everyone says, "We had a bad feeling," but nobody can point to who said it and what they said and why it got ignored.

And then there's the human part nobody likes talking about: incentives.

If you reward speed and only speed, you'll get speed. You'll also get corner-cutting, even from good people. If you punish the person who raises a problem, you'll get silence. If preparedness is treated as an obstacle, teams will route around it. If it becomes a partner that helps them ship safely and quickly, they'll show up early and ask for help before the launch blog post is drafted.

I keep thinking about the professor and the recommendation letter, too. It was such a small error. It didn't change a product roadmap. It didn't trigger a policy. It just made a student feel processed.

That's the damage that makes me nervous, because it's quiet and cumulative. It's also easy to rationalize. "We're busy." "It was an honest mistake." "It's not that serious." All of that can be true, and the trust still disappears.

I don't have a perfect ending for this. I don't even have a comforting one.

If you build powerful tools and you ship them quickly, you will hurt people in small ways you didn't intend. That's not a dramatic prediction. It's a boring one. The question is whether you notice early, whether you can stop the bleeding without pretending it's not happening, and whether you're willing to slow down sometimes even when slowing down feels like losing.

Some days you'll do it well. Some days you won't. And a lot of the time, if you're doing it right, nobody will clap. They'll just keep using the tool and never know how close you got to shipping something worse.