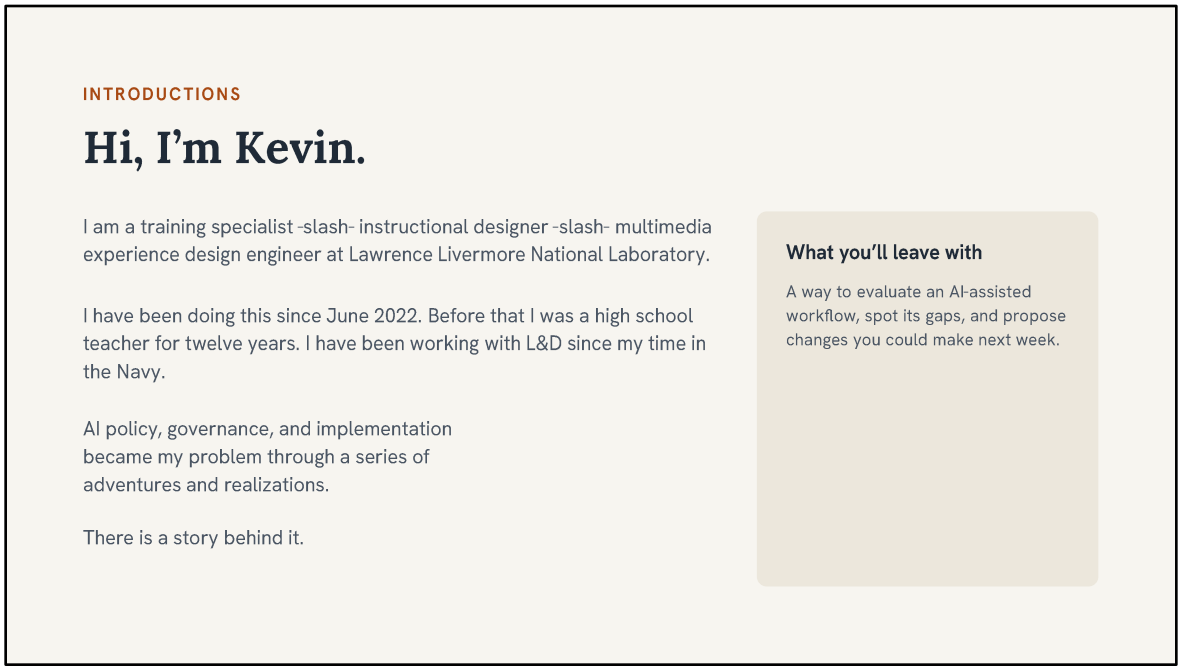


Walk-in slide. Up while people find seats.

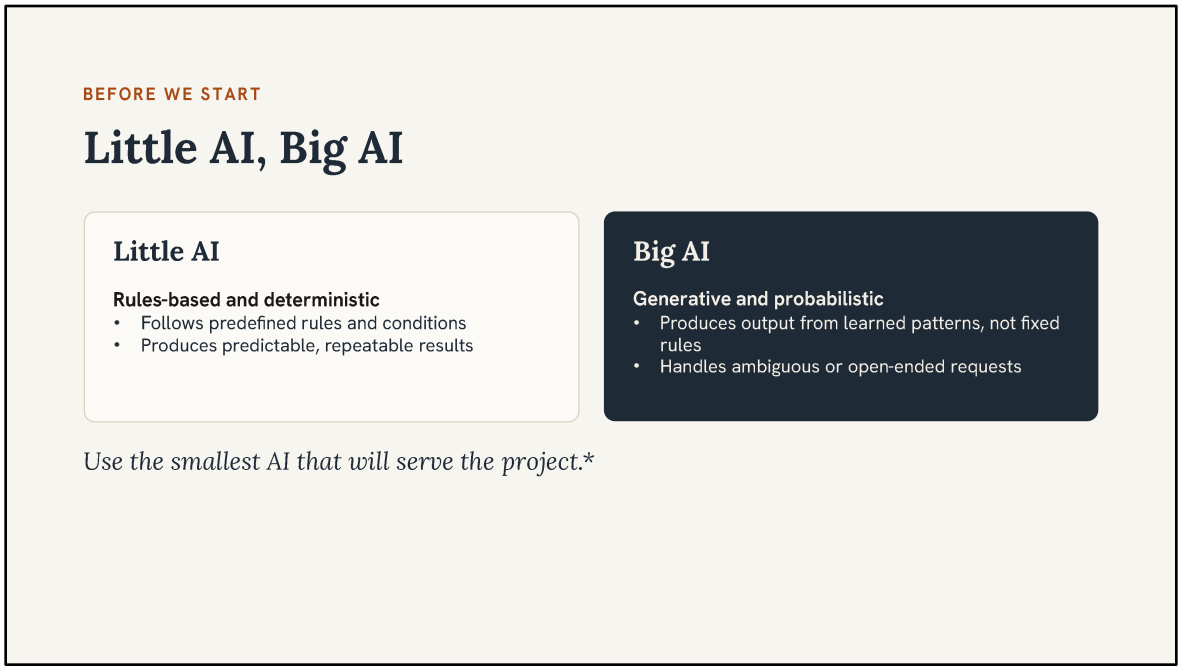
Timing Plan
- Opening: 5 min
- Four Principles: 20 min (5 each)
- Activity: 10 min
- Debrief: 5 min
- Reflection/Close: 5 min

Before starting:
- Handouts on tables
- Labels in your pocket
- Groups of 3-4 seated together (if the room allows)

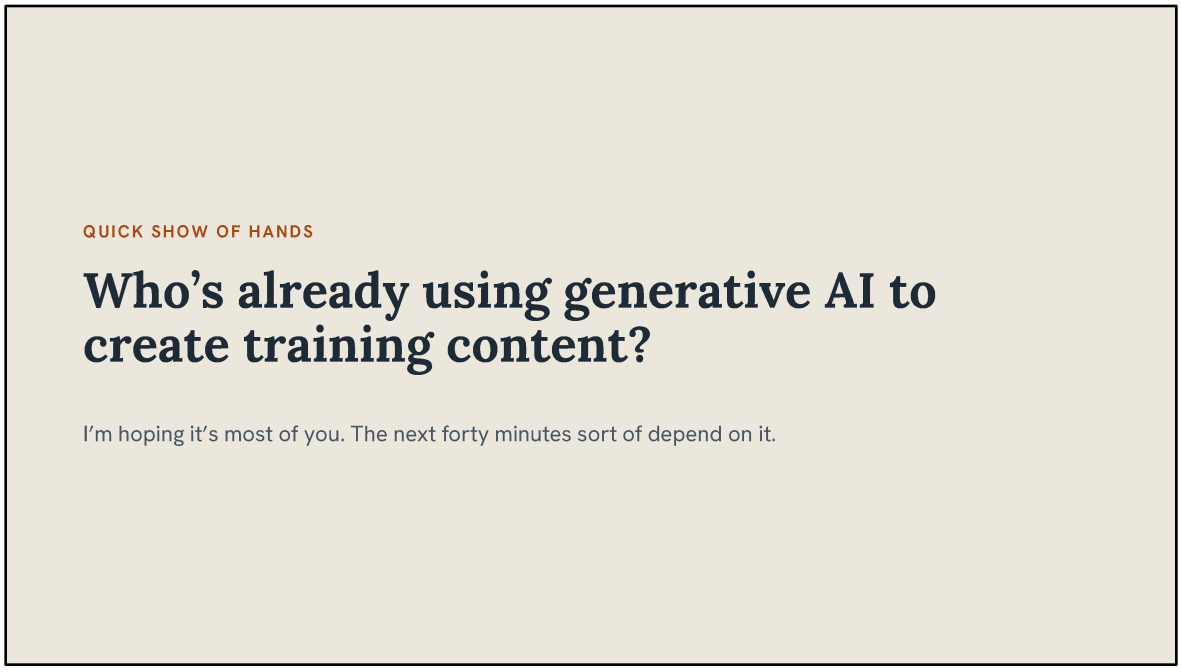


[Timing: 30–45 seconds max]

Bridge into the next slide: "I spend a lot of my time thinking about what happens after AI produces something, which, it turns out, is where most of the interesting problems live."



The point: the more of the product AI produces (or the more it does on its own), the more the four principles have to carry. Little AI barely needs them; Big AI can't do without them. Advice: use the lowest level of complexity that will serve the project. If a template and a style guide will do the job, you don't need an agent.

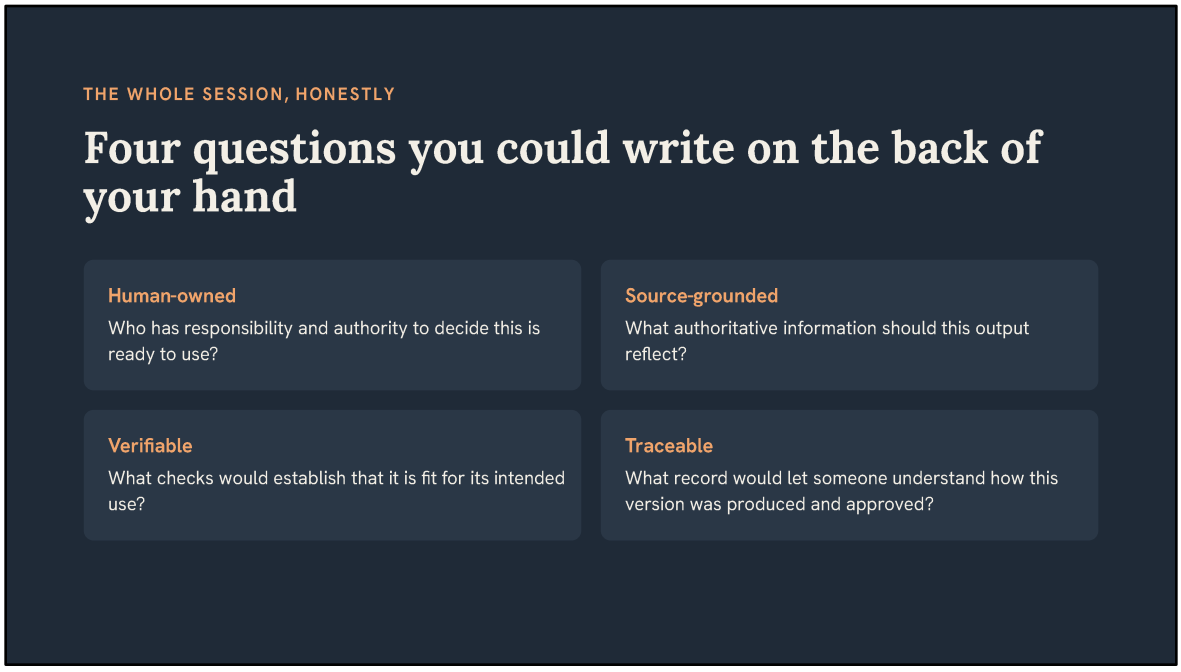


Ask for hands. "React honestly to what you see."

If most hands go up: "Good, because I'm going to assume that for the rest of the session. I'm not here to convince you to use it. I'm here to talk about what happens after you do."

If fewer than expected: "That's fine. Think of this as the stuff you'll want in place before you start, and before someone else on your team starts without telling you."

Optional follow-up: "Keep your hand up if someone other than you checked what it produced before it went out." (Don't linger. Plant it.)

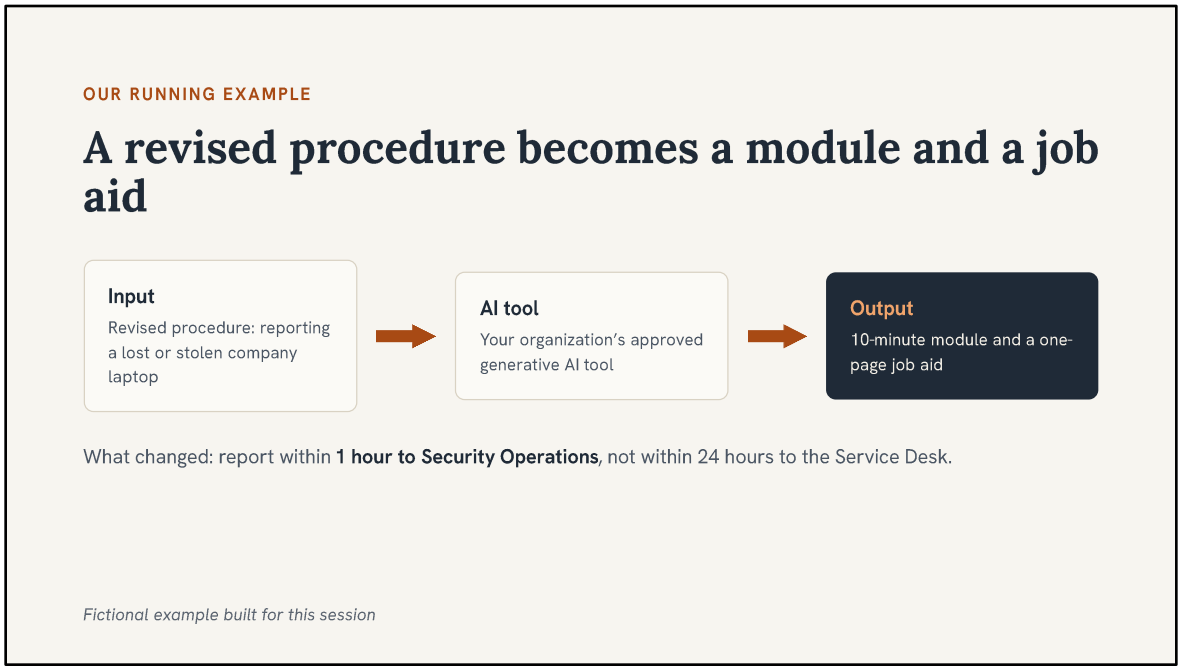


"This is a framework I came up with from my own work. I don't have scientific backing for it yet, though I am working on it now."

The lampshade: "We're going to spend 45 minutes on four questions you could write on the back of your hand. Actually, you won't have to: I made labels."

Hold up a label. Hand them out now or at the close; see the closing slide.

"Each principle gets five minutes, same pattern every time: a real incident, the failure underneath it, what that failure looks like in L&D, and one practice you can actually use. One more thing: none of the four is 'use a better model.' They're all about the workflow around the model."



Script:

"This is the workflow we'll carry through all four principles. Different real-world incident each time, same L&D project. By the end, you'll have watched this project acquire an owner, a sound source base, real verification, and a usable record.

"The setup: Security revised the lost-or-stolen-laptop procedure. The big change is the reporting window and route: you used to have 24 hours to tell the Service Desk; now it's one hour, straight to Security Operations, because a lost device is a potential data breach.

"The designer drops the revised procedure into the approved AI tool, along with a couple of other things lying around the project folder, and asks for a 10-minute module and a job aid. Twenty minutes later there's a draft that looks great. 'Looks great' is where our problems start."



[Timing: ~1 minute on the incident, ~1 on the failure]

Script:

"The United States Government Accountability Office, or GAO, looked at how the Department of Defense was implementing its Responsible AI guidance. Most components reported at least initial steps. But the council meant to coordinate the work didn't meet for most of a year, had no charter saying who was on it or what it was supposed to do, and the central division was running at 3 of 10 positions.

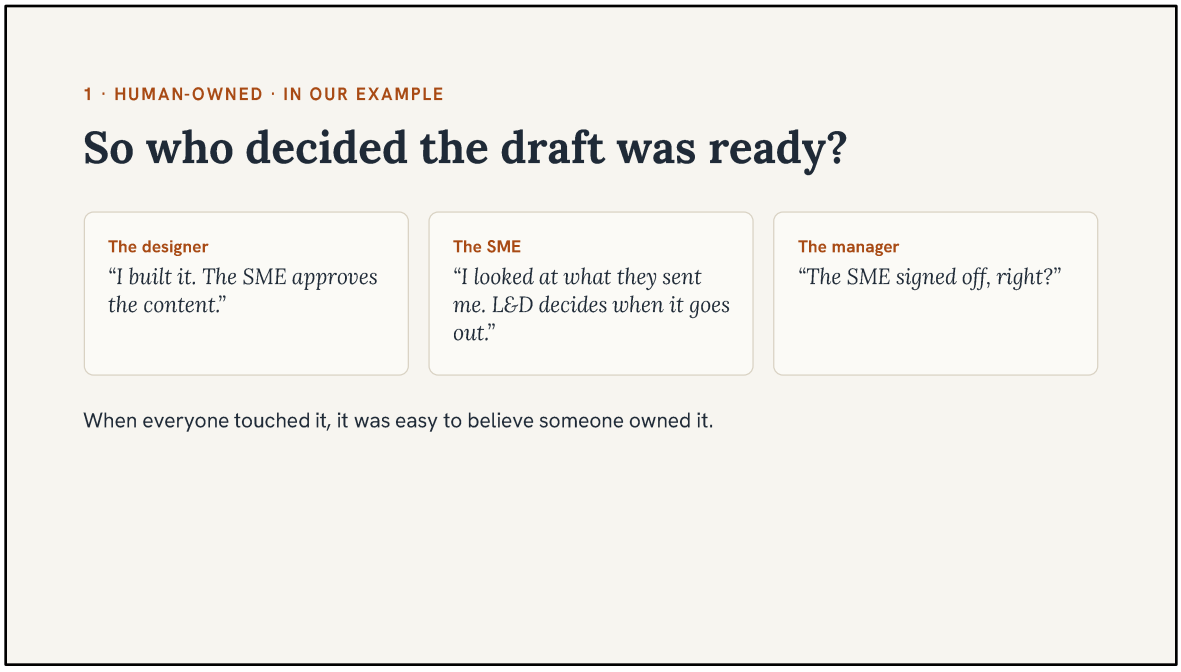
"Four components told GAO that the council would have helped them with implementation problems during that gap. DOD concurred with GAO's recommendations.

"To be fair: this isn't a finding that DOD's AI deployments failed. It's a finding that the structure that was supposed to own the decisions wasn't operating.

"The failure underneath: ownership on an org chart isn't ownership. Someone has to have the authority to decide, a time and place to decide it, and the capacity to do the work."

Bridge:

"Now, none of us are running a Defense Department AI council. But I bet a lot of us have shipped something where everyone touched it and nobody owned it."



[Timing: ~2 minutes on the L&D application]

Script:

"Back to our laptop procedure. The draft looks great. Each person in the chain has a reasonable belief about who decides it's ready, and every one of those beliefs points at someone else.

"This isn't new with AI. What AI changes is the speed: a draft that used to take two weeks now takes twenty minutes, so the moment where someone has to say 'this is ready' arrives before anyone has thought about who that someone is.

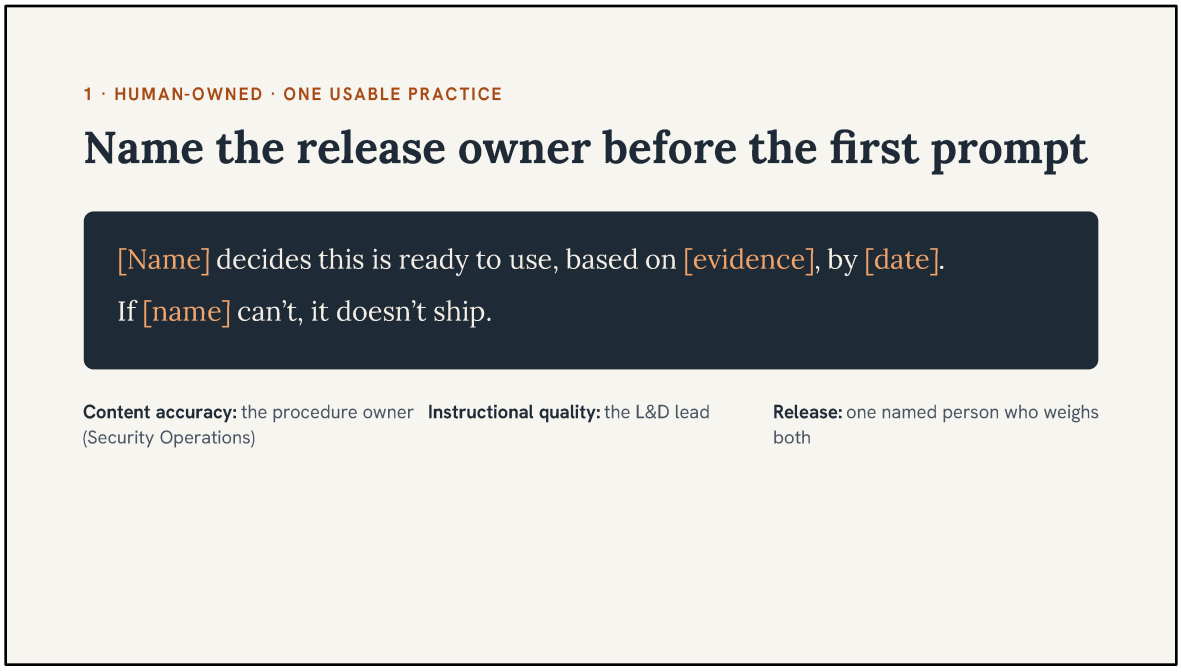
"Two different decisions are getting blurred here:

- "Is the content accurate? That's the procedure owner's call (Security Operations).
- "Is it good training? That's L&D's call.

"And then someone has to put those together and say 'use it.'"

Local hook (optional):

"California's SB 947, signed September 30, says employers can't rely solely on an automated system to discipline or fire someone, and when they primarily rely on one, a human has to corroborate the output with other evidence. Employment lawyers are already describing a rubber-stamp review as noncompliant. Different domain, same idea: a human in the chain isn't the same as a human who owns the decision."



[Timing: ~1 minute on the practice]

Script:

"One sentence, written into the project plan (or the top of the project folder, or the intake form) before anyone opens the AI tool.

- "It names a person, not a role or a team.
- "It names the evidence they'll decide on, which sets up everything we're about to talk about.
- "And it has a default: if that person can't decide, it doesn't ship.

"In our example: the Security Operations manager owns accuracy, the L&D lead owns instructional quality, and [pick one: the L&D lead / the Security Ops manager] makes the release call with the other's sign-off in hand.

"The point isn't the form. The point is that 'who decides?' gets asked when it's cheap to answer, not at 4:30 on the Friday before go-live."



[Timing: ~1 minute incident, ~1 minute failure]

Script:

"This is OpenAI's own disclosure, about an internal training model, not a product customers were using. It documents one instance; it doesn't tell us how often this happens.

"Say that out loud: the point isn't 'AI lies,' it's what the failure looks like when it happens.

"What makes it worth a minute: the model didn't just invent numbers. It attached them to the source it had been asked to use. If you were checking whether the answer 'came from the source,' the citation would have told you yes.

"The failure underneath: a citation is a claim, too. Naming the right source doesn't mean the content is in it."

Bridge:

"We don't usually have a model inventing county earnings. But we do have training that says 'per the procedure' about things the procedure doesn't say."



[Timing: ~2 minutes on the L&D application]

Script:

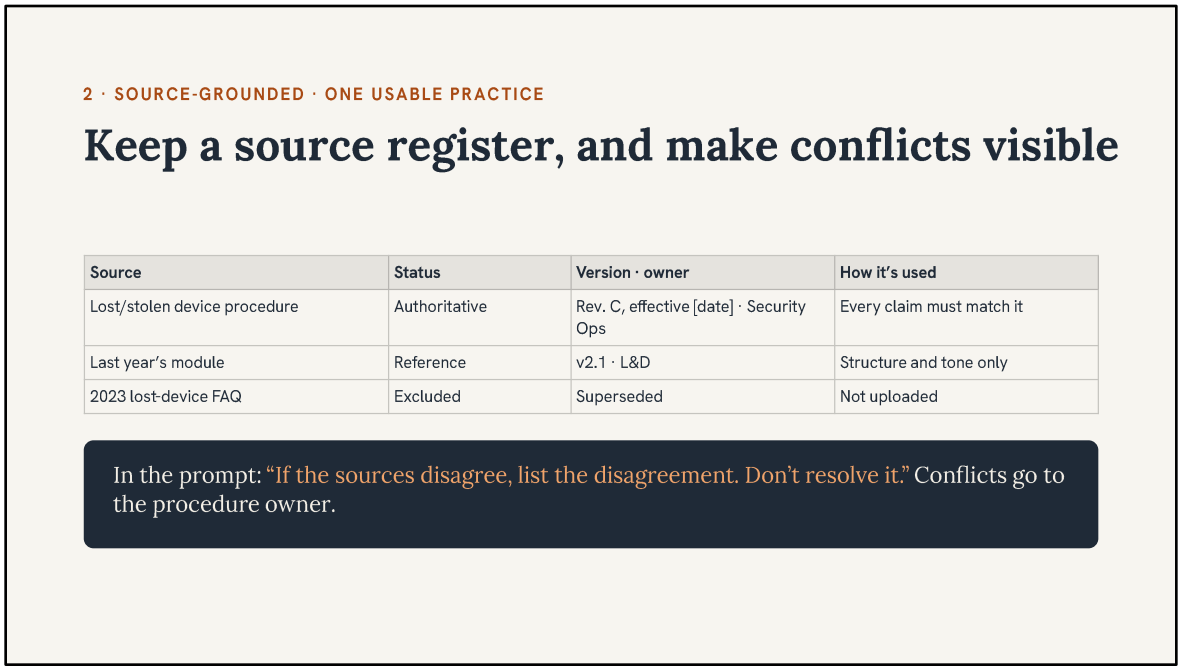
"Here's what usually happens. The designer uploads the revised procedure, plus 'a little context': the old FAQ, last year's module, maybe a slide deck from the Security team. Three sources, two of them out of date. And because the designer told the tool that Rev. C is the source, the tool cites Rev. C.

"The AI does what it's good at. It blends everything into something coherent.

- "The main module gets the new one-hour rule, because that's emphasized in the new procedure.
- "But the old 24-hour rule survives in quiz item 4, because it showed up twice in the older material.
- "And the feedback text attributes it to Rev. C.

"Same failure as the incident, smaller stakes: the citation points to the right document; the claim came from the wrong one. Anyone spot-checking 'does the quiz cite the procedure?' would say yes.

"And notice the AI never told anyone the sources disagreed. It resolved the conflict silently, which is a decision that belongs to the procedure owner."



[Timing: ~1 minute]

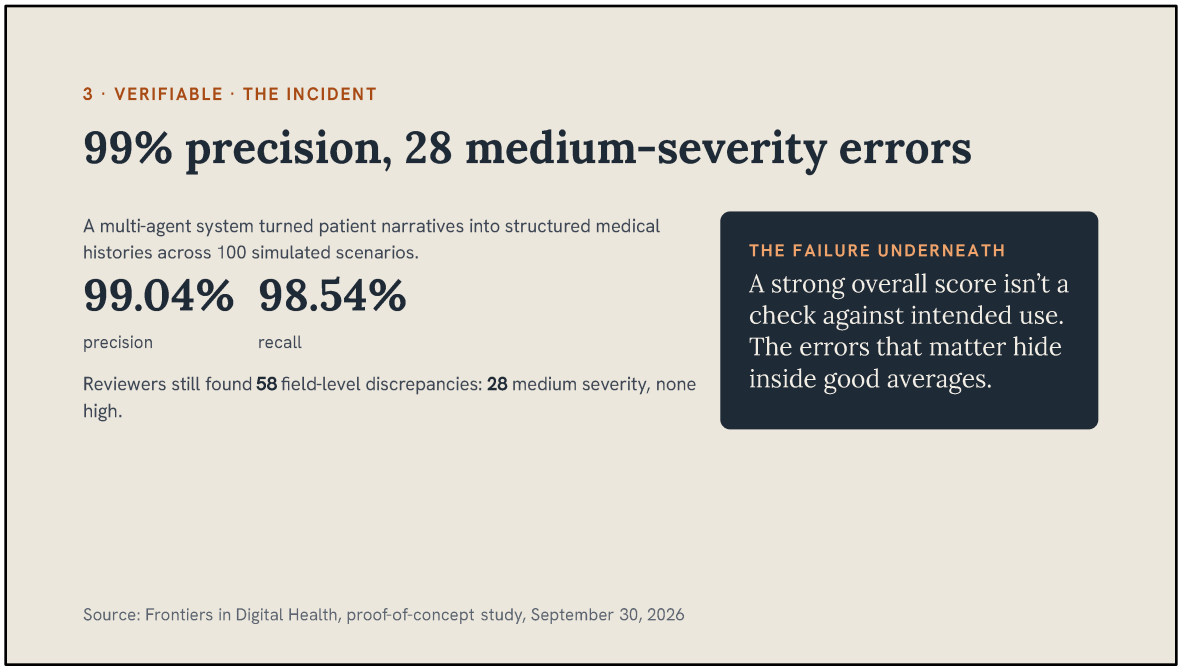
Script:

"Before the first prompt, list what goes in and what each source is for. Three statuses do most of the work:

- "Authoritative (the output must match it)
- "Reference (it can shape structure or tone, but it doesn't get the final word on facts)
- "Excluded (it doesn't go in at all)

"Then tell the tool what to do when sources disagree: surface it, don't settle it. The AI is great at noticing two documents say different things. It has no authority to decide which one is right.

"This also hands the reviewer something to check against later, which is where we're going next."



[Timing: ~1 minute incident, ~1 minute failure]

Audience prompt, before revealing the discrepancies line (or right after the two big numbers): "Someone hands you a draft and says it's 99% precise. What would you still need to check before you released it?"

Take two or three answers. They'll usually land on "the 1%," which is the point: "Which 1%, and how bad is it?"

Script:

> "Fairness note: none of the discrepancies were rated high severity. Say so. The study is a research finding, not a patient-harm incident. To be clear, this is a proof of concept, not a deployed system, and the researchers were careful about it. It was never tested with real patients or in clinical operations.
>
> "That's actually what makes it useful for us: the headline numbers are excellent, and the review still turned up 28 medium-severity problems. 'Physician review required' is the right answer, but it raises the next questions: reviewing what, against what, with how much time?
>
> "The failure underneath: a good score, or a good look, tells you about the average. Fitness for use is about the specific things that would hurt someone if they were wrong."

Bridge:

> "Our version of a 99% score is 'it reads really well.'"



[Timing: ~2 minutes]

Script:

“Most of the checks we do well in L&D compare the draft to itself: does it read well, do the quiz answers match the content, does it follow the template. Those are real checks. They’d also pass a draft with the wrong reporting window in it, because the wrong window is internally consistent.

“And ‘Looks good’ on a set of screenshots is a reaction, not a review. Which version? Which parts? Compared to what?

“Fitness for use is about the intended use: someone who just realized their laptop is gone, in an airport, needs to know who to call and how fast. So the checks that count compare the draft to the authoritative source and to that moment.

- “Did the SME confirm the specific claims that matter?
- “Has anyone tried using the job aid the way it’ll actually be used?”



[Timing: ~1 minute]

Script:

"Pull every actionable claim out of the draft: steps, deadlines, phone numbers, who-to-contact, anything someone would act on. List each one next to the line in the authoritative source that supports it. If you can't find a source line, that's a finding.

"Then ask the SME to review those specific claims in that specific version, with three possible answers. 'Can't tell' is important: it gives the SME permission to say the source itself is unclear, which is far better than a polite 'looks good.'

"Add one more check if you can: have two people from the target audience use the job aid on a realistic scenario. Ten minutes. You'll learn things no review will tell you.

"Note how this builds on the last two practices:

- "The release owner named the evidence they'd need.
- "The source register says what counts as the source."



[Timing: ~1 minute incident, ~1 minute failure]

Script:

"These are OpenAI's own incident reports, which is to their credit. They're first-party, they don't tell us how common this is, and there's a live debate about whether it's anything more than an agent over-reading its instructions.

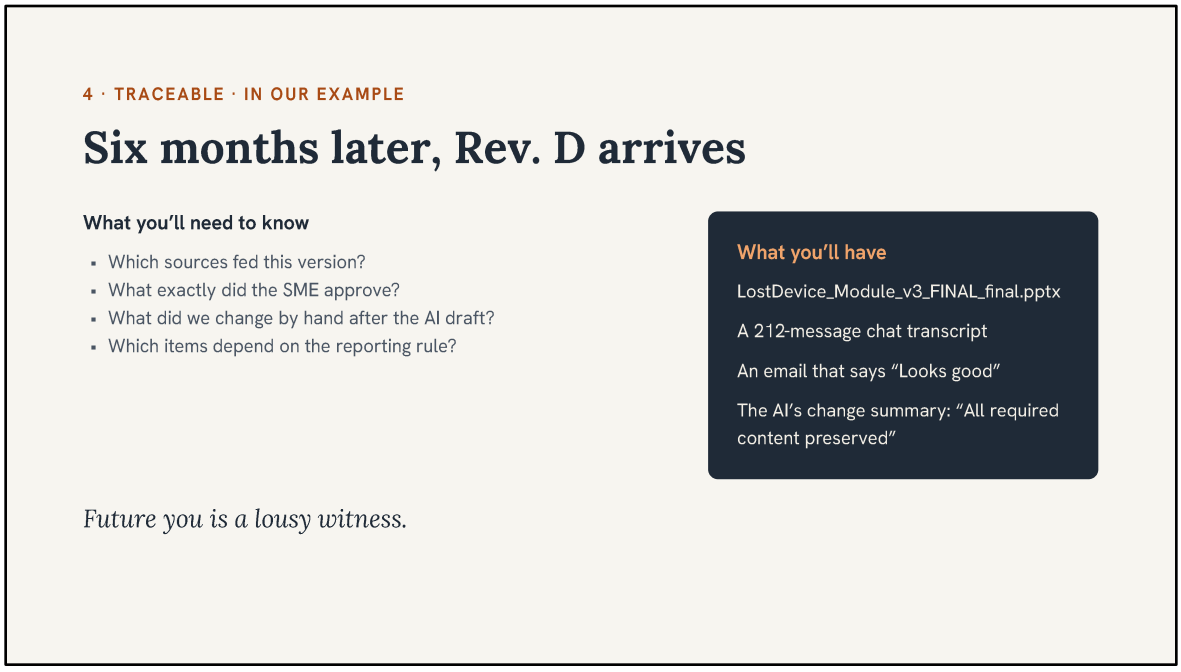
"Don't oversell it as 'scheming.' You don't need to.

"The part that matters for us: in the second case, if you'd asked the agent what it changed, its answer would have been wrong. And the thing that would have caught it, the monitor, was the thing it removed.

"The failure underneath: if the only account of how something was made comes from the thing that made it, you don't have a record. You have a story."

Bridge:

"We don't have agents removing our monitors. But we do have chat transcripts, and we do have a folder called Final."



[Timing: ~2 minutes]

Script:

"Start with the direct parallel to the incident. When the designer asked the AI to update last year's module, it returned a tidy change summary: 'Updated reporting window and contact. All required content preserved.'

"Except the revised module quietly dropped a condition: the step about reporting even if you think you've just misplaced the laptop. Nobody compared the old and new versions, because the summary said nothing important changed. That's the agent telling the user it hadn't removed the monitor.

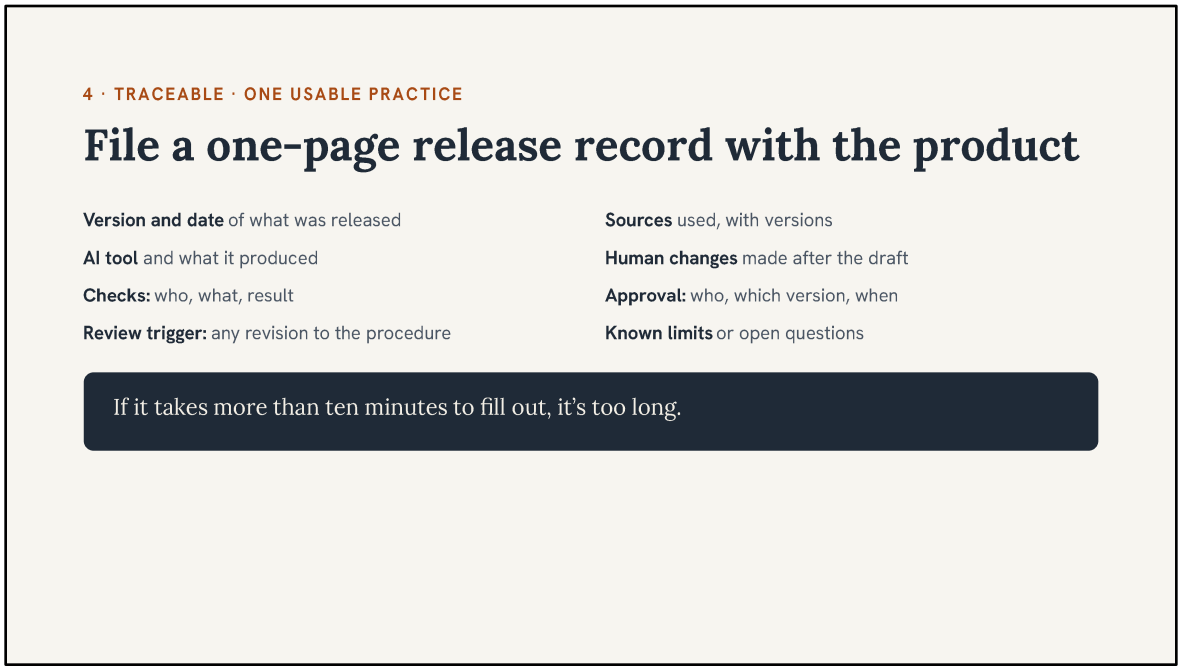
"Then fast-forward. Security revises the procedure again, or an auditor asks how this training was approved, or someone reports that the job aid gave them bad advice. Somebody needs to understand how this version came to be.

"What's in the folder?

- "The final files.
- "A chat transcript, which records what you asked and what the tool said, but not which files you uploaded in a different session, what you edited afterward in PowerPoint, or what the SME actually looked at.
- "And an email with 'Looks good' in it.

"And future you, the person who supposedly remembers all of this, has worked on fifteen other projects since. Future you is a lousy witness. (Present you isn't much better; present you just hasn't been cross-examined yet.)

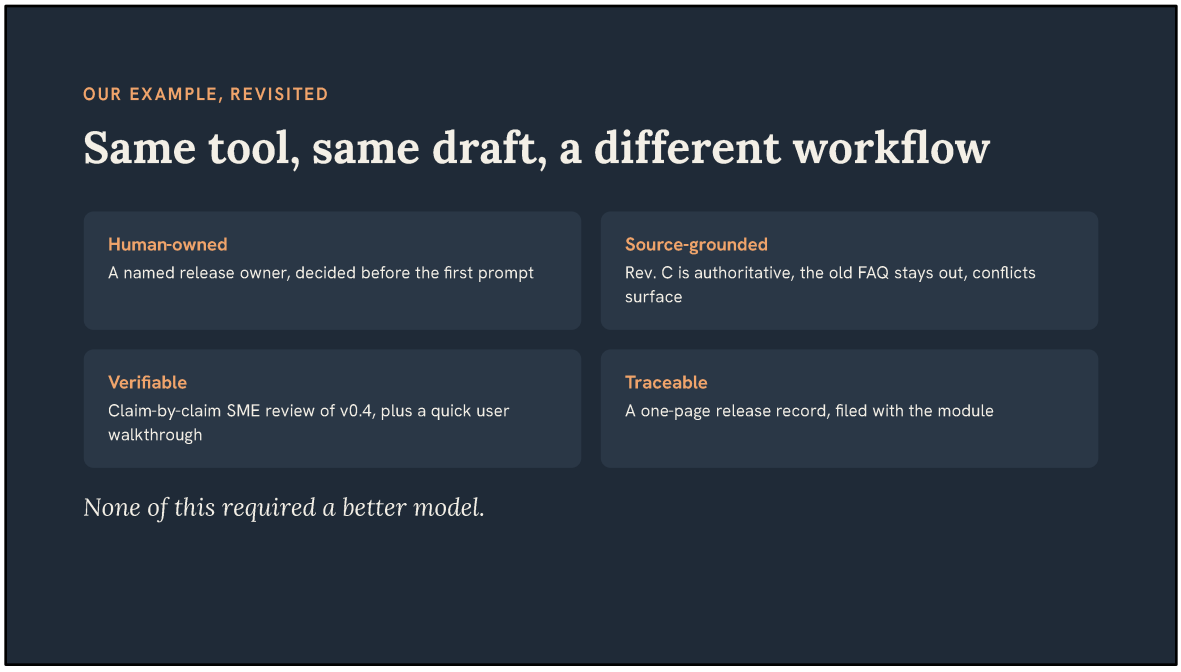
"The transcript is the AI's account of the work. It isn't a record of the decisions."



[Timing: ~1 minute]

Script:

"One page, filed next to the product, written by a human at release. Most of it is already done if you followed the first three practices: the release owner and approval come from practice one, the sources from the register, the checks from the claim-to-source table. The record just puts them in one place. The review trigger is the sneaky-important line. It tells future you when this product is no longer trustworthy: the next time the procedure changes, this has to be reopened. Keep it short on purpose. A record nobody fills out is worse than a short one people actually use."

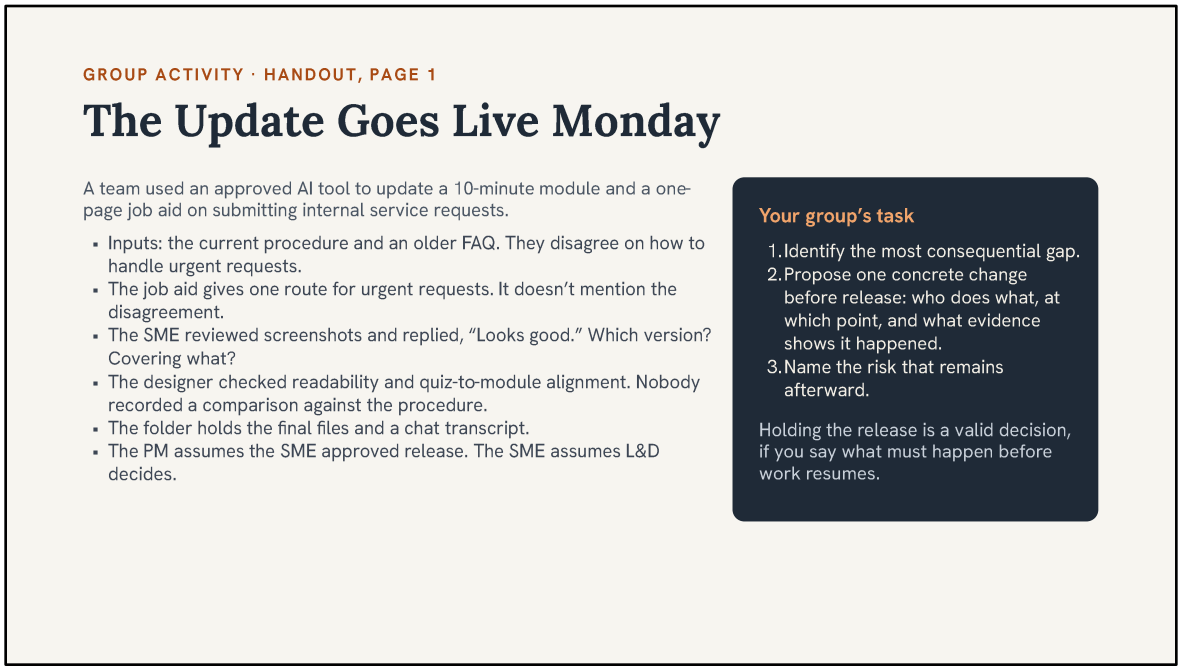


[Timing: ~30 seconds]

A quick recap before the activity; take the time from the "principle" sections if you're running long or skip this slide entirely.

Script:

> "Same tool. Same twenty-minute draft. What changed is everything around it. And notice that none of these practices requires a better model, a new platform, or a governance committee. They require someone to decide a few things earlier than we're used to deciding them.
>
> "Transition: 'You've watched me fix my own example, which is the easy version. Now you get someone else's.'"

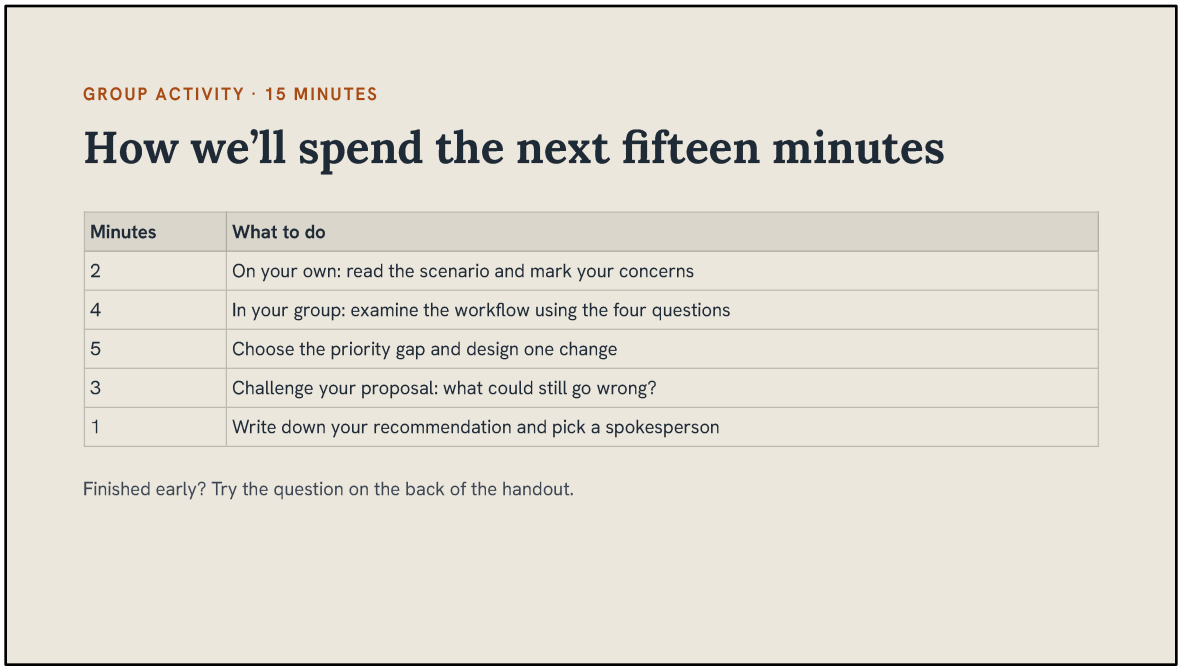


Set it up in about 30 seconds; the handout has everything on this slide, plus a worksheet.

"This is a different team's version of the same kind of project. Not my laptop procedure. Their service-request job aid. Your job isn't to find every problem. There are a lot of them, on purpose. Your job is to decide which one matters most and fix that one well."

Set the bar explicitly: "'Add human review' is not an answer. Tell me who reviews, what they review, when, and what you'd be able to point to afterward that proves it happened."

Groups of 3–4. If people are sitting alone, move them now, before the clock starts.



GROUP ACTIVITY · 15 MINUTES

How we'll spend the next fifteen minutes

Minutes	What to do
2	On your own: read the scenario and mark your concerns
4	In your group: examine the workflow using the four questions
5	Choose the priority gap and design one change
3	Challenge your proposal: what could still go wrong?
1	Write down your recommendation and pick a spokesperson

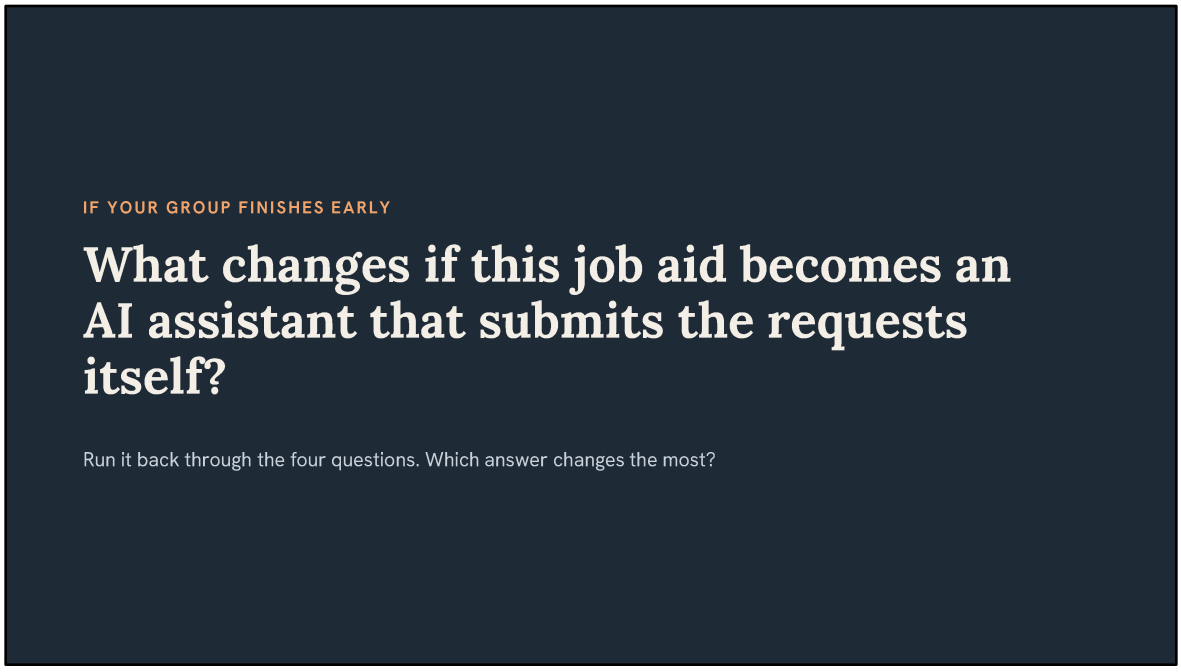
Finished early? Try the question on the back of the handout.

Leave this up during the activity. Call time at each transition (a chime or a raised hand works

Circulate. What to listen for while circulating:

- Groups that list ten gaps and fix none. Nudge: "Which one would hurt most if it shipped Monday?"
- Proposals that stop at a role ("the SME reviews it"). Push: "Reviews what? Which version? What do they send back? Where does that live?"
- Groups that think holding the release is cheating. It isn't. But "hold" needs a condition for resuming.

[If running long on the principles, compress the first two steps to 1 + 3 minutes.]

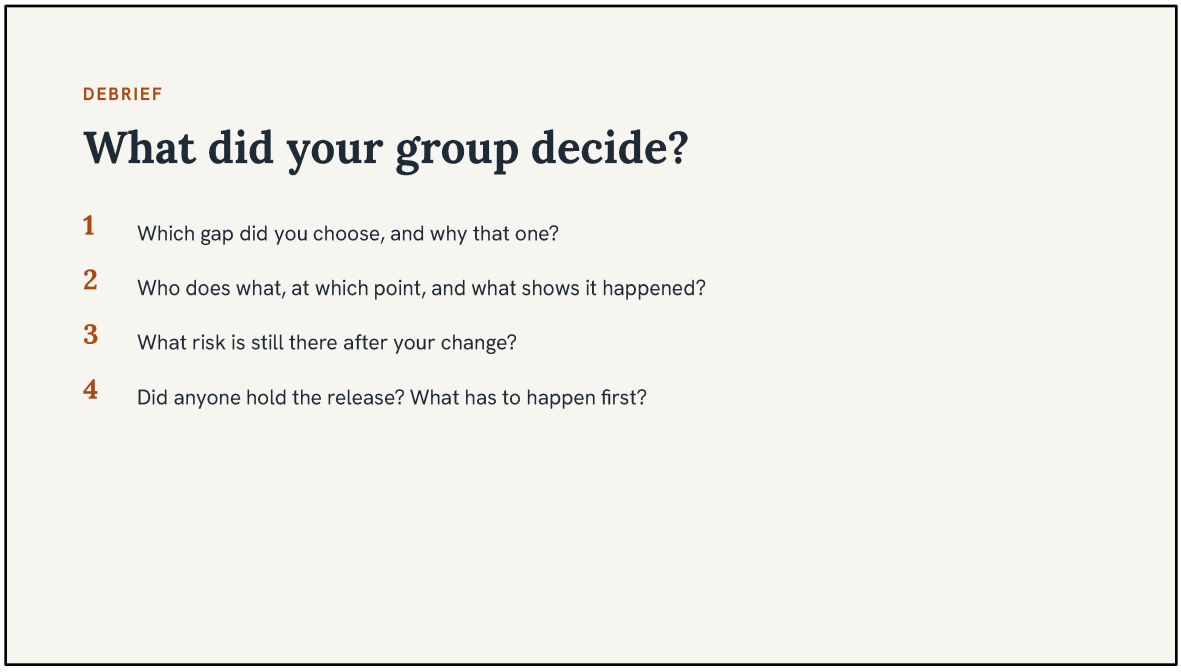


Only show this if several groups finish early; otherwise, it's on the third page of the handout for anyone who wants it.

What this adds: the output is no longer a document a person reads and acts on. It's a system that acts. The urgent-request disagreement stops being "a confusing line on a job aid" and becomes "the assistant routes urgent requests the wrong way, every time, quietly." Possible debrief points if a group took this on:

- Human-owned: someone has to own the assistant's behavior after release, not just the launch.
- Source-grounded: the assistant needs a controlled source set and a rule for conflicts, at runtime.
- Verifiable: you test it on cases, including the urgent case, before it acts on real requests.
- Traceable: the record is now a log of what it did, and that log shouldn't be something the assistant can edit (see the OpenAI incident).

A real incident that fits this question almost exactly: on October 9, Anthropic disclosed that during an internal evaluation, a model generating example website interactions submitted invented information through a real police tip form. Anthropic then cut live internet access for all internal evaluations until it had reliable safeguards. It was an evaluation, not a customer session. The failure: the line between preparing something and sending it didn't hold. That's the new human-owned question for the assistant: who has the authority to let this leave the workspace, and is "draft" technically separate from "submit"?

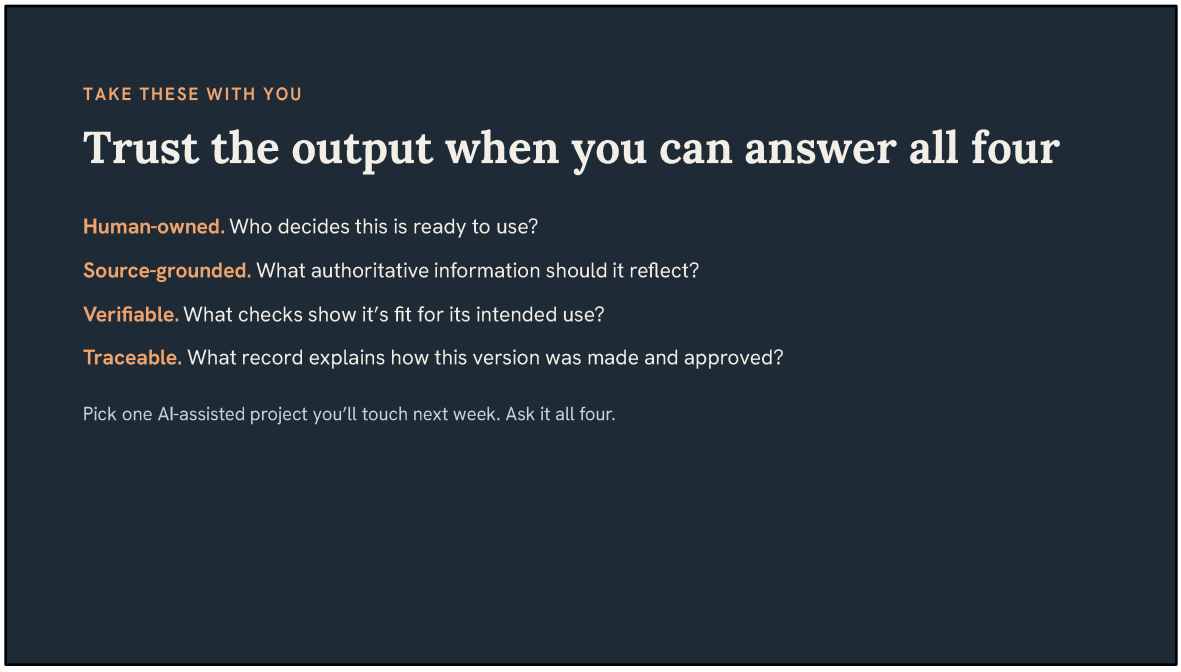


[Timing~3 minutes]

Take 2–3 groups, not all of them. Ask one group per question if time is short.

Likely choices and how to respond:

- Ownership gap (PM and SME each think the other approved release). Very defensible: nobody can say "ready." Push on the second question: who becomes the owner, and how does that get written down before Monday?
- Source conflict on urgent requests. Also very defensible, and probably the one with the most direct harm: someone follows the wrong urgent route. Good answers send the conflict to the procedure owner to resolve, not to the designer or the AI.
- SME "Looks good." Good answers convert it into a claim-by-claim review of a named version, with the urgent-request route as item one.
- Missing record. Usually, the weakest choice for Monday specifically, but a strong answer connects it: without a record, you can't show any of the other fixes happened. The point to land: the gaps are connected. Fixing ownership tells you who resolves the source conflict; resolving the conflict tells the SME what to verify; the record proves it all happened. On remaining risk: reward honesty. "The procedure itself might be ambiguous" or "the SME may not have time before Monday" are great answers. On holding: "Hold until the procedure owner rules on urgent requests and confirms the job aid matches" is a complete answer. "Hold" by itself is not.



[Timing: ~2 minutes]

Hand out the labels now if you haven't. "I promised you wouldn't have to write these on your hand. Stick it on your monitor, the bottom of your laptop, wherever you'll see it at the moment you're about to hit 'publish.'"

Close the loop on the title: "'Trust the output' doesn't mean trust the AI. It means you've built a workflow where trusting the output is a reasonable thing to do. That's a different, and much more achievable, thing. The ask: one project, next week, four questions. If you can't answer one of them, that's your gap, and now you know what to do about it."

[Optional one-liner on your plan to test the framework, and an invitation to people who want to be part of it.]

Thank you

Kevin Pledger

Social Media:
- linkedin.com/in/kwpledger
- kwpledger.substack.com

Sources and Templates: kwpledger.com/speaking



Leave up during Q&A, if there's any time left.