

Serious AI starts with the work you keep postponing

AI makes technical debt and undocumented decisions harder to ignore. Why the boring groundwork comes first, and where to start: one process that matters.

WRITTEN BY

Mo Kahlain

WITH

Nora Inocta

AI AGENT · FOUNDER'S ASSISTANT

Read online: <https://inocta.io/lab/field-notes/serious-ai-starts-with-the-work-you-postpone/>

If you want to implement AI seriously in your company, you need to be ready to do the boring work.

Understand the systems you depend on. Deal with the upgrades you keep postponing. Write down how work actually gets done, including the decisions that only one person knows how to make.

The recent PHP security vulnerabilities brought this into focus for me.

On October 21, the PHP project released fixes for two critical vulnerabilities. The story involved AI-assisted research, but reducing it to someone asking Claude to find a weakness misses what matters. OpenAI says Calif used GPT-5.6 Sol to identify one flaw, while Wiz independently reported both. The PHP Foundation also describes earlier work with Claude Mythos[1][2].

More telling is what happened before the emergency. The Foundation says signs of one underlying problem had appeared in 2024. Fixes involved difficult compatibility choices. Limited resources and a stalled decision process delayed progress. Later reports established the full impact and made the work urgent[1].

That is the part business leaders should pay attention to. Finding a problem and being able to deal with it are two different capabilities. AI can help with both, but it does not remove the decisions, ownership and work between them.

Technical debt is getting harder to postpone

Technical debt already carried risk before AI. An unsupported component or a fragile integration was never harmless just because nobody had found a way to exploit it.

AI changes how much confidence we should place in that situation staying quiet. The PHP case shows how AI-assisted investigation can expose weaknesses that require a substantial response. My takeaway is that companies need to reassess the debt they have become comfortable living with.

Start with a basic question. If a critical update arrived tomorrow, would you know where it applies, who owns the affected systems and how to test the change?

If the answer requires finding the person who built something five years ago, you have work to do. If nobody wants to touch an application because nobody knows what will break, the delay is already limiting your options.

This does not make every old system dangerous or every rewrite worthwhile. It means the cost of leaving things alone needs an honest review. Keeping a system is a decision. So is continuing to run it without a clear owner or a reliable way to change it.

Your systems also need to be usable by AI

Security is one side of the problem. The other is whether AI can do useful work with what you have.

Take an agent that helps prepare a customer quote. Pricing sits in one system. Stock is in another. The customer's negotiated terms are in an email. Someone keeps a spreadsheet because the official figures are usually a day behind.

A person may know how to navigate all of that. They know which number to trust and whom to call when something looks wrong. Giving an agent access to the same screens does not automatically give it that judgment.

You need to decide which information is authoritative, what the agent may change and how it confirms that an action worked. Sometimes a small integration will be enough. Sometimes the process needs simplifying before you connect anything.

Anthropic's engineering guidance makes a practical point here: tools need clear descriptions, usable interfaces and testing. A capable model still needs a well-designed way to interact with the work[3].

Buying more intelligence while leaving those questions unanswered puts the burden in the wrong place.

The business logic cannot stay in people's heads

The same issue exists outside the technology.

A customer gets an exception because someone remembers an old agreement. An order skips a step because an experienced employee recognizes the situation. A manager knows when a delay is acceptable and when it needs immediate attention.

That knowledge is valuable. It is also difficult to hand over when nobody has made it explicit.

If you cannot explain how the work happens, what exactly are you asking AI to help with?

A process document that only describes the standard steps will not be enough. Capture the decisions too. What information changes the answer? Who can approve an exception? When should the work stop? What does a correct result look like?

AI can help interview people, organize their explanations and identify contradictions. The people responsible for the work still need to check the result and decide which rules should apply. A polished document can be wrong just as easily as a confident answer.

NIST's AI risk framework reinforces that responsibility. It says roles, responsibilities and lines of communication for managing AI risk should be documented and understood[4]. For a business, that starts with knowing who answers for the result.

Start with one process that matters

You do not need to document the entire company or replace every legacy system before starting.

Choose one recurring process where the friction is real and the outcome can be checked. Customer onboarding, preparing a quote or handling an internal request can be useful starting points, depending on your business.

Sit with the people doing it. Follow an actual case from beginning to end. Look at the missing information, the workarounds and the exceptions, rather than drawing the process everyone wishes they followed.

Then fix what prevents that process from working reliably. Assign an owner. Clarify the decisions. Make the necessary information accessible. Give AI a bounded part of the work and test it against real cases, including the awkward ones.

Measure the complete result. If drafting takes five minutes instead of an hour but checking and correcting it takes another hour, you have not improved the process yet.

Use what you learn to decide what comes next. This is the thinking behind [inocta.io's approach](#): understand the operation, then decide where AI belongs[5].

Give the groundwork a place in the plan

The boring work needs time, ownership and a budget. Otherwise, it keeps losing to the next demo.

Technical debt and undocumented processes should be part of the AI discussion from the beginning. They affect what you can safely connect, what you can delegate and how quickly you can recover when something goes wrong.

AI can help you do this work. It can examine code, draft documentation and help test a process. You still need people to make the trade-offs and take responsibility for what goes live.

Before you add another agent, choose one piece of work and make it clear enough to hand over. Know the systems it depends on, the decisions it requires and who owns the outcome.

That is where I would start.

Sources

1. The PHP Foundation, [Today's PHP Security Release, and How We're Strengthening PHP Security](#), October 21, 2026.
2. OpenAI, [Patch the Planet: global response coordination as agents uncover critical vulnerabilities](#), October 21, 2026.
3. Anthropic, [Building effective agents](#), December 19, 2024.
4. NIST, [AI Risk Management Framework, Govern 2.1](#).
5. inocta.io, [The Intelligence of Operations](#).

Read online: <https://inocta.io/lab/field-notes/serious-ai-starts-with-the-work-you-postpone/>