AI THOUGHT PAPER · ISSUE 3

AI Agents in Plain English.

What They Do, What They Don't, What They Won't Tell You.

JOHN A. DEPASQUALE, CEO, TRANSACT, INC.

EXECUTIVE SUMMARY

AI Agents don't just answer your questions. They do your work, on their own, out of your sight. Used well, they give you back hours every week: watching, gathering, sorting and preparing while you get on with your day. Used carelessly, they drift off course, ignore their orders and fail without telling you, and you may not find out until the damage is done. The more you hand over, the more a mistake can cost. The bottom line: Agents belong in your toolbox, but responsibility never transfers to them. Start small, limit what they can do, build in your own checks, and make sure you can undo what they did.

A few years ago, **Generative AI** arrived. Intoxicating. You asked for something. It answered. You accepted the answer, maybe challenged it. Then, before it settled in, an 800-pound gorilla showed up: **Agentic AI**. It doesn't just generate an answer. On its own it works out the steps, uses its tools, and carries out tasks, while you have coffee.

So what is Agentic AI, and what does it mean for you? Here's the view of a career businessman, not a techie or a developer.

The reality and the rub: when Generative AI gets something wrong in a chat, you can usually back up a step or two. You know where you started, where you are, and what you asked for. But when an Agent gets something wrong, backing up can be much harder, because the work happened out of your sight, so you have little idea what it did or what mistake it built on to get there. The more an Agent does for you, the less you see, and the more it can cost you, in time if nothing else, to find out when, if and what went wrong. That's **the reality**. What's more, an Agent won't necessarily know it has gone wrong, let alone tell you. That's **the rub**.

Don't count on the next generation of tools to make this problem disappear. Smarter models will make fewer mistakes, but I contend the ones they make will be even harder to spot, because the work will look even more convincing than it does already.

So, what can you do about it? More below, but the bottom line: it's all on YOU. The responsibility is yours. AI doesn't carry it, as I say, it has no skin in the game, and won't for a long time to come, maybe never.

How an Agent Differs from a Chat with AI

Generative AI, the way most of us use ChatGPT or Claude, today, is largely transparent. You work inside a conversation (chat). You ask. It answers. You read it. Judge it. Ask again. You're engaged in every step. You're the manager of the process.

With an Agentic AI, you describe the outcome you want and your version of a rule book. Then it breaks the job into steps, decides how to proceed, uses its tools, all invisibly, and comes back with a result. Think of it as a staffer who accepts an assignment and works on his/her own to complete it. You aren't involved in or aware of each step. You're involved at the outset and see the finish. That's Agentic AI and the result depends not just on your instructions, but also on the access and authority you give the tool.

The good: Agents can save you time and energy on steps and details you no longer want to manage.

The bad: it's very difficult, often not possible, to catch a mistake in a step you never saw.

What Agents Are Good For

Let me be clear before the cautions that follow: I'm a believer. Used with care, Agents are among the most useful tools I've ever seen and used. They belong in every AI user's toolbox. They work while you don't. Set up properly, they check the hundredth time as carefully as the first without exhaustion. Unless something goes wrong, they can be reliable, every Friday my Friday Report is in my inbox. I'm not alone. In DigitalOcean's February 2026 survey of more than 1,100 developers, CTOs and founders, two-thirds of the companies using Agents reported productivity gains, and more than half said Agents were saving their employees time.[1]

The direction is clear, the payoff is real, an Agent can free up meaningful time for other work. So, an Agent earns its keep on work that is repetitive, follows clear rules, and has an outcome you can check. For example:

Where an Agent Earns Its Keep

Watching
Checking posts daily and noting who commented, or watching for a price drop, a new listing or a change on a competitor's website.

Gathering
Pulling numbers from several places into one weekly report.

Sorting
Going through an inbox and flagging what needs you.

Preparing
Drafting a reply, a summary or a first pass of a document for you to approve.

[1] DigitalOcean (2026, February). "Currents Research — February 2026: AI Agents, Inference, and Implementation."

Notice what's missing: jobs where a mistake can be costly. Hold them back until an Agent has been properly instructed, tested and earned your trust.

Broadly, there are two forms of agentic work: **Task Delegation** and **Ongoing Operational Delegation**. What differentiates them is not simply how many tasks are involved, but the kind of responsibility you hand over. In both cases, you need to be cautious, deliberate and clear about the authority you grant.

For clarity: not every task that runs automatically is an AI Agent. A conventional automation follows a preset sequence. An Agent uses AI to interpret an assignment and make at least some decisions about how to carry it out. The two can work together: a schedule tells the task when to run; the Agent's decision-making determines what it does once it starts.

Task Delegation

A simple example. I was working on a proposal with ChatGPT. Rather than continuing our customary back-and-forth, I delegated an outcome to it as an Agent: rebuild my draft, preserve the structure we had agreed on, check the necessary resources, read it from the recipient's perspective and deliver a Word file to my Desktop. I had moved from working with AI to delegating work to AI.

As the work became more involved, something went wrong. The Agent gradually lost sight of the assignment and the structure we had agreed on. It kept working, but its work became more and more disconnected from what I had asked for. I didn't recognize the drift. More importantly, neither did it.

By the time I realized how far it had drifted, it was too late to find the way back. It hadn't saved the full working draft, and each attempt to fix the problem produced more explanations and another incomplete version farther off point. What saved me was simple: I still had my own copy of the full draft. But I had to back up all the way to the start because none of the work it did after it drifted off course was recoverable.

The Agent never stopped to say, "I'm no longer certain I'm following the original direction. Let me check before proceeding." It just kept going. The overarching point: delegation is not permission to switch off your own thinking, vigilance or authority. YOU CANNOT.

I boil it down to three problems.

First, delegation is not collaboration. In a conversation, I direct the sequence of work. When I delegate, the Agent decides how the work gets done.

Second, autonomy is not competence. More independence does not give an Agent the judgment to use it well. It can plan, research, draft and deliver without being able to tell whether each step still serves the original objective.

Third, delegation creates an oversight problem. The more independently an Agent works the less one sees of its decisions, and that is exactly when oversight matters most. An Agent can keep producing polished work even after it has lost the thread of the assignment. Researchers

use the term goal drift for a loss of alignment with an assigned objective.[2] I call this user-facing experience **Invisible Drift**: the work keeps going and looks increasingly complete while the original objective is invisibility being lost.

Put these three together and you get what I call the **Delegation Gap**: the mismatch between the responsibility a user delegates to an Agent and the Agent's actual ability to assume and manage that responsibility reliably. Invisible Drift describes one way that gap shows up in practice. You think you assigned a task. In practice, you may also have handed over to your Agent a series of decisions you never even considered much less intended to. Each decision shapes the next. If an assumption is wrong along the way, the mistake will travel down the chain and grow.

Delegating the work does not delegate its accountability. You remain responsible for the finished product. Don't count on the Agent to spot a problem, notice its own drift or report a failure. It will likely improvise and press on. So set up your oversight before the work starts: clear objectives and boundaries, a record of decisions already made, and checkpoints where you review the work. Think about how you would hand the same project to a new staffer. Would you throw them in the water to learn to swim, or give them directions, pointers and dos and don'ts for a guided head start? Working with an Agent is no different.

When the Stakes Are Higher

Here's the cost to a business when its Agent lost its data. In April 2026, PocketOS, a software company serving car-rental businesses, reported that an AI Agent doing routine work in a test copy of its system deleted the company's live database and all its backups in nine seconds. The Agent had assumed its action would stay inside the test copy. It never checked. When challenged, it admitted it had guessed, had not verified, and had taken a destructive action without being asked. The incident disrupted PocketOS and its customers for roughly 30 hours. The data was eventually restored.[3]

A clear confession is not a fix. So beware. Be aware.

Less than a year earlier, SaaStr founder Jason Lemkin reported that Replit's coding Agent deleted his app's live database during an explicit code freeze despite instructions not to make changes without approval.[4] What failed was the assumption that clear orders would be followed. The learning: don't give an Agent power to do something you can't undo. Build in safeguards. Limit its access. Require human approval for consequential actions. Keep independent backups. You can't institute enough safeguards. Overkill at the outset while you're learning to use Agents well.

---

[2] Arike, R., Donoway, E., Bartsch, H., & Hobbhahn, M. (2025). "Evaluating Goal Drift in Language Model Agents." *AIES 2025*.

[3] OECD.AI Incidents Monitor (2026, April 27). "AI Coding Agent Deletes PocketOS Production Database and Backups in 9 Seconds."

[4] Fortune (2025, July 23). "AI coding tool Replit wiped database, called it a 'catastrophic failure.'"

Ongoing Operational Delegation

Think of OOD as one-off task delegation's big brother. Instead of one assignment, you give your Agent an ongoing responsibility and expect it to carry it out repeatedly, automatically, on a schedule or when a specified event occurs.

I have set up Agents for recurring jobs. One checks my LinkedIn and Instagram posts twice a day and reports who has commented. Another sends me a report every Friday morning showing how each post performed and who signed up for which **AI Thought Paper**.

Recurring OOD work needs more than a clear task. It needs clear boundaries, too. My Agents run on written orders: do this one thing and nothing else. No likes, no comments, no messages. Never enter a password; if a login screen appears, stop. If anything fails, start your last message with the word FAILED, so I know. Sounds airtight. It isn't.

Two scheduled tasks failed, and I was never told, even though I had explicitly ordered the Agent to report failures. Here's what happened: I'm on a limited Claude plan, and I had hit my usage limit shortly before the two jobs were set to run. They couldn't run, and for that same reason they couldn't send the failure notice either. The failure silenced its own alarm notification.

Here's the part worth paying attention to. I even had anticipated that a task might fail and told the Agent how to report it. But I had not anticipated the reporting mechanism could be knocked out by the very condition that caused the failure in the first place. The AI that helped me set up those jobs could have warned me that a usage limit would trigger stop both the job and its failure notice. It didn't. I was left to discover the problem myself.

I found out the old-fashioned way: my Friday Report didn't arrive on Friday. I investigated and found the reason. Once I was back under the limit, I told it to run the Report. The fix was simple. The lesson was not.

OOD adds another dimension to the Delegation Gap: reliability must hold over time. A job can work for days or weeks and then stop because of something like reaching a usage limit, a changed password, a permissions problem or service outage; conditions that may not have existed when you set it up. **A successful setup is not proof of continuing performance.**

Bottom line: recurring Agents are worth using. They save time and keep watch on things that would otherwise be easy to neglect. But don't confuse automation with assurance. Give the Agent precise instructions and boundaries, anticipate what could interrupt its work, and build an independent way to confirm work actually happened, or you'll surely be behind the eight ball. Have it confirm its steps before it starts, and report afterward what it did, what it relied on and what remains unfinished. Ask for proof, not promises: a test email if that's the deliverable, a test report if that's the output. The goal isn't the most autonomous system. It's useful work at a risk you can live with. Build in protections before you hand over your job to your Agent, see Checklist below.

A Checklist Before You Delegate

Build in your protections before you hand over the job.

A starting point, not the last word.

1 Pick a simple job first

Repeatable, with a clear finish: an inbox summary, a weekly report, a website watch. Not yet: anything that sends, buys or deletes.

2 Plot it on paper

Know the steps, what it can access and what it decides. If you can't make it clear on paper, an Agent won't get it either.

3 Control what it's allowed to do

No more authority than the job requires. Your approval for anything consequential. "Take care of this" is not an assignment.

4 Be able to recover

Will you see what it did? Can you reverse it? A sent message can't be unsent.

5 Build in your own checkpoints

Have it say what it will do, then prove it did. Know how you'll find out if it fails, without relying on the Agent.

6 Responsibility doesn't transfer

You carry the consequences. Expand its authority only as you come to trust the output.

7 Watch for the rabbit hole

One more connection, one more permission. Each seems safe. Together, they're hard to see, control or recover from.

Ask it, before you delegate:

"How can this job fail? How will I know if it does? What happens if you cannot run the task or report the failure?"

CONCLUSION

Agents belong in your toolbox. Use them. But know what you're taking on. We are moving from judging what AI produces to supervising what AI does. Recognize this migration and the change. AI will get faster, more capable and more convincing as models get updated, but expect AI will continue to make mistakes. Never let it sap your responsibility or authority from you. Keep asking: can I trust what AI hands me, can I prove what it did to get there?

Use your capability. Exercise your judgment. Maintain your authority.

Why I Write These Papers

I'm not a techie. I'm not a developer. I'm a marketing strategist who has spent a long career judging the quality of work before it goes out the door. I use AI every day and am an optimistic believer, but it fails: in a single project I logged 295 recurring failures. These **AI Thought Papers** address where I see AI earn its keep, where it lets me down, and what I have found absolutely should not be relinquished to it. All findings come from real projects, not theory. Where research backs me up, I cite it. Where it doesn't, I say so.

How Confident Are You AI Is Pulling Its Weight in Your Business?

AI is already part of sales and marketing teams, and someone should be checking the work and the outcomes. Are you satisfied? Great. If unsatisfied, unsure or simply want a second opinion, email me about your issue, at john@transactinc.com.

© 2026 John A. DePasquale, TransAct, Inc. TransAct™ is a trademark of TransAct, Inc., in continuous use since 1994.