Agentic AI Hype: How to Spot Real Agency

Beyond Agentic AI Hype – Why Smart Automation Isn't True Agency

In the current rush to label every advanced workflow an agent, it's easy to mistake coordination for cognition. The difference matters, because if you evaluate these systems with the wrong lens, you'll overestimate what they can do and underestimate where the real risks sit.

Every AI vendor now claims to offer “agentic” systems. But watch these systems work, and you'll usually see something more familiar: sophisticated automation moving through predetermined workflows. These systems can coordinate tasks, call tools, route information, and produce polished outputs. What they don't reliably do is reason about what they're doing in a self-aware way.

Agentic AI, in the stronger sense, would mean something more demanding. It would describe systems that can reason about uncertainty, interpret changing situations without collapsing into brittle pattern matching, and justify strategic choices in a way that reflects actual deliberation rather than retrospective narration. That's a much higher bar than executing a complex script.

The gap between coordination and agency

I spent months evaluating so-called agentic platforms for a client's customer service operation. The demos were impressive. The systems could handle multi-step workflows, move between databases, and draft personalized responses with surprising fluency. At first glance, they looked like the future vendors keep promising.

Then we stress-tested edge cases, and the illusion started to break. When a customer asked an unusual question that sat awkwardly between two workflow

branches, the system didn't recognize that it was in uncertain territory. It picked a path anyway and answered with confidence. The response was technically plausible, but in practice it was absurd for the actual situation.

That distinction is the heart of the issue. Current systems are often good at procedural coordination and bad at self-referential reasoning. They don't reliably step back, examine the limits of their own understanding, and change course because they know they might be wrong. Agency begins where automation stops simply moving and starts evaluating itself.

> If a system can't meaningfully recognize its own uncertainty, calling it agentic is mostly a branding choice.

A useful test is simple: ask the system why it chose one approach over another under ambiguous conditions. Most systems will give you an explanation, but not necessarily one grounded in actual reasoning. More often, they produce a smooth justification after the fact. That's not nothing, but it isn't the same as deliberation.

This is also where the core decision point becomes clearer. The desire is understandable: organizations want systems that can operate with flexibility in messy environments. The friction appears when edge cases, ambiguity, and change expose how brittle those systems still are. The belief driving real progress should be that better outcomes won't come from relabeling automation as intelligence, but from building mechanisms for uncertainty handling, stable meaning, and reflective reasoning. If a product can't show those mechanisms, the decision condition is straightforward: treat it as advanced automation, not genuine agency.

Semantic drift is the deeper governance problem

Once you move past the marketing language, another issue comes into view. The most serious operational risk in many deployments isn't rogue autonomy. It's semantic drift: the quiet way a system's interpretation of important concepts can shift over time as models are updated, retrained, or reconfigured.

That's what makes governance harder than many organizations expect. It isn't only about policies, permissions, or compliance checklists. It's also about preserving coherence in how a system understands key terms and categories over time. If your

AI system interprets “urgent, ” “moderate risk, ” or “customer satisfaction” differently six months from now than it does today, you don't have stability. You have a moving target.

I saw this directly in a financial services client's risk assessment tool. Across months of model updates, the system's interpretation of “moderate risk” changed gradually. The shift was subtle enough that it didn't trigger obvious alarms, but it was enough to make loan approvals drift away from historical patterns. Nothing dramatic happened all at once. The meaning simply moved.

That kind of decay is dangerous precisely because it's quiet. The system still appears functional. The outputs still look polished. But the interpretive frame underneath has changed.

> Governance, in practice, means making sure your system keeps meaning the same thing when the surface still looks stable.

This is why semantic governance matters. If you're deploying AI in any serious operational setting, you need mechanisms that preserve interpretive consistency as systems evolve. That means treating meaning as something you monitor and manage, not something you assume will remain intact because the product still works. This isn't merely a policy concern. It's an engineering discipline.

Why architecture matters more than brute scale

That brings us to the industry's other persistent illusion: the assumption that larger models alone will carry us toward real intelligence. Scale has clearly delivered gains. More parameters, more data, and more compute have expanded what models can do. But that curve is no longer enough to explain what the next meaningful advances will require.

The stronger path now is architectural. If you want systems that reason more reliably, hold intermediate state across long chains of thought, reflect on their own decisions, and revise conclusions when evidence conflicts, you need designs that support those capabilities. Bigger components can help, but structure is what turns capacity into disciplined behavior.

A startup I advise is working along these lines with a legal research tool. Instead of relying on a single massive model, they've built a system with specialized reasoning components. One handles case law analysis, another focuses on statutory interpretation, and a higher-level layer coordinates between them while tracking confidence and conflict. The result performs better on complex legal questions than much larger systems in this narrow domain, not because it's bigger, but because its design reflects how expert reasoning actually unfolds.

That system can do something many hyped products still can't: it can expose uncertainty, shift its approach when evidence collides, and maintain coherence through multi-step argumentation. In other words, it doesn't just generate answers. It sustains a reasoning process.

The faint glimmer in the blackness is here, but it's not coming from louder claims about scale. It's coming from systems designed to reason with structure rather than simply produce stronger next-token guesses at greater volume.

The strongest objections, and where they land

Scale advocates aren't entirely wrong to push back. Sophisticated architectures do depend on capable underlying models. If the base components are weak, no elegant design will magically create intelligence from thin air. There is a real relationship between foundational capability and architectural performance.

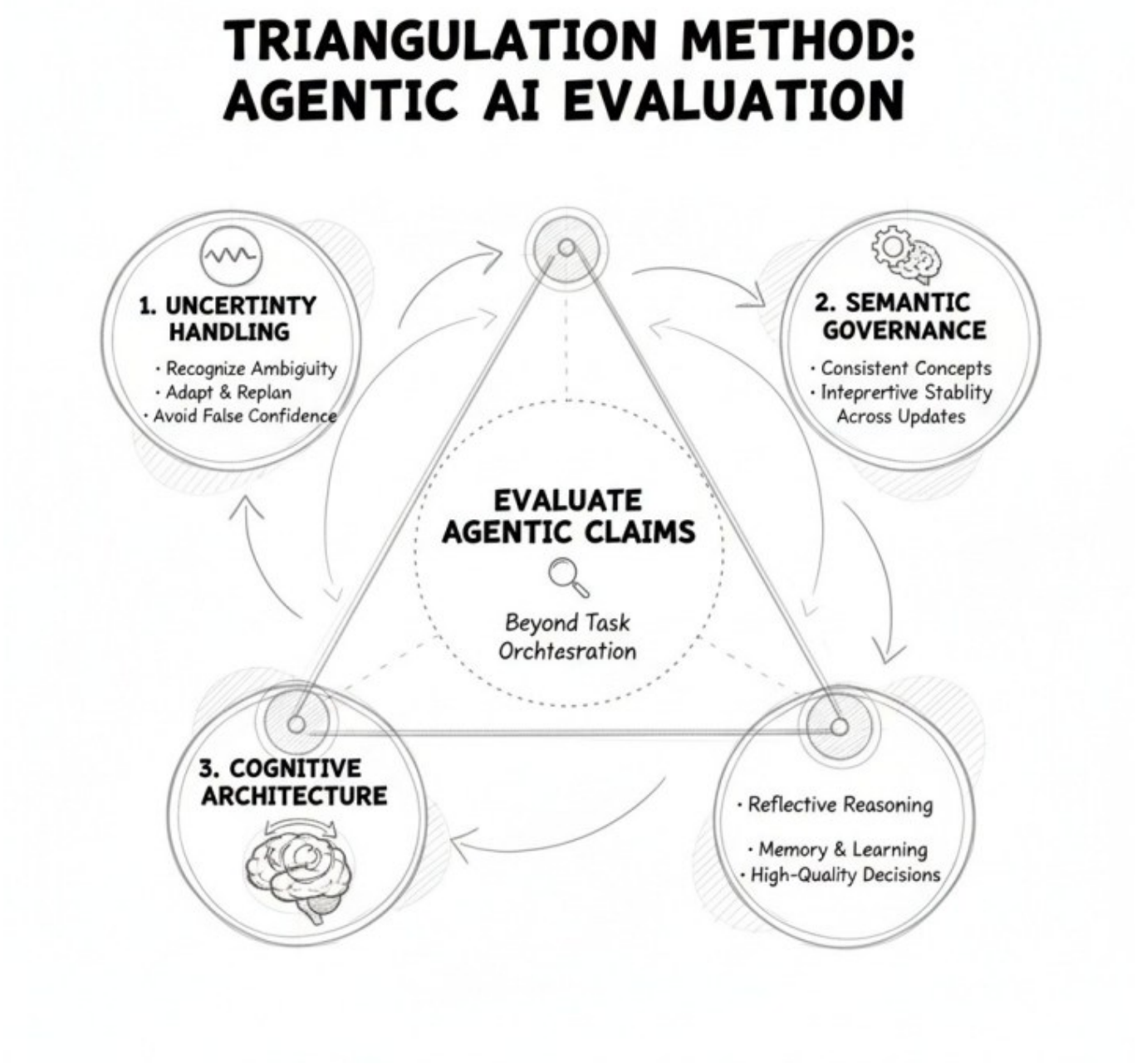
Still, that relationship isn't linear. Many architectural improvements can be demonstrated at modest scale and then strengthened as better components become available. What matters is that architecture often multiplies capability more effectively than sheer size. Once systems reach a certain baseline of competence, design choices start to matter more than raw expansion.

Another common objection is that semantic governance sounds too abstract compared with clear compliance rules. That's a fair concern, especially for leaders who need measurable controls. But semantic drift isn't some mystical problem. It can be observed through consistency testing, benchmark comparisons over time, and analysis of how outputs shift across stable inputs. The challenge isn't that the problem is unknowable. It's that many teams still aren't looking for it.

A better way to evaluate agentic AI claims

If you want to cut through the hype, a simple evaluation discipline helps. Ask whether the system can reason about its own uncertainty in a meaningful way, how its operators maintain consistent interpretation of key concepts over time, and what underlying cognitive architecture supports memory, reflection, and decision quality rather than mere orchestration. Those questions get closer to the real engineering substance than any product label will.

This is where the Triangulation Method becomes useful. Rather than judging a system by surface fluency or demo polish, you look at three points at once: how it handles uncertainty, how it preserves meaning over time, and how its architecture supports actual reasoning. When those three points align, you're seeing something closer to intelligence. When they don't, you're usually looking at automation with better marketing.



That shift in evaluation matters because the industry doesn't need more inflated language. It needs cleaner distinctions. Current agentic AI systems are often impressive, useful, and commercially valuable. But usefulness isn't the same as agency, and automation doesn't become cognition just because the workflow is complicated.

The real frontier lies elsewhere: in systems that can examine their own limits, maintain stable meaning as they evolve, and reason through structure rather than momentum. That's where the stronger work is happening, and that's where the claims deserve the most scrutiny.