What makes agentic software testing trustworthy?

Rethinking governance, delegation, and human oversight in the age of autonomous testing



UiPath

Content

3 Why trust matters more as AI becomes more autonomous

4 Trust must be designed into the system

6 UiPath AI Trust Layer: Putting guardrails around AI

7 Delegation without giving up control

8 From human-in-the-loop to expertise-in-the-loop

9 Humans should handle judgment and exceptions, not every action

10 Autonomy and control can coexist



Why trust matters more as AI becomes more autonomous

For most of the recent AI era, trust has come down to a familiar question: can we rely on the answer? Generative AI gives you something to consider, like a summary, a test case or a piece of code. The AI suggests, and you decide what happens next.

Agentic AI changes that relationship.

Once AI can pursue goals, pick its own tools, work across systems and take action, the quality of its output is only part of the trust question. The bigger question is whether you're comfortable letting AI act on your behalf. In software testing, that can mean going from AI that helps write tests or analyze results to agents that play an active role across the whole testing lifecycle. Moving from AI-assisted to agent-led testing is really a shift in where agency sits.

Concerns about autonomous AI make sense. You can catch a bad recommendation and reject it before anything happens. An agent that can act largely independently is a different story. The more it can reach into enterprise systems, data and processes, the more you need to understand the boundaries of the autonomy it's been given. That's where the conversation about trust must broaden.

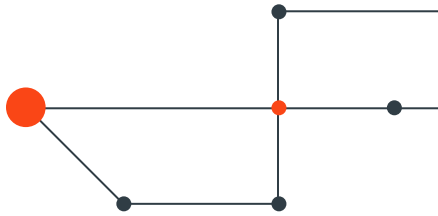
You need to know that an AI system is capable, and that its access and authority fit the job. Once an agent starts working autonomously, you also need to see what it's doing. If you can't observe, trace or make sense of its actions after the fact, autonomy gets very hard to govern.

Even well-designed agents will run into situations that are ambiguous, unfamiliar or outside what they were set up to handle. So trust also depends on what happens at the edges of an agent's or automation's competence or authority. Just because an agent can take the next step doesn't mean it should. A trustworthy agentic system is designed to recognize when it has hit a boundary, when it needs more context and when a decision belongs to someone else.

> **"The real shift happens when you connect these islands of intelligence into governed autonomous flows."**
>
> **Ingo Philipp,**
> **VP, Product Management, UiPath**

Governed autonomy sits between two extremes. At one end, a person must approve every meaningful action. You keep control, but you lose much of what makes autonomy worth having. At the other end, AI gets broad freedom with too few boundaries and too little oversight. Enterprises need neither.

Why trust matters more as AI becomes more autonomous (continued)

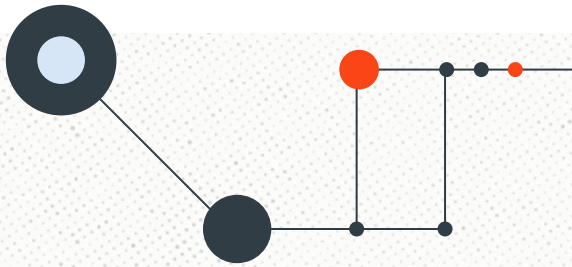


The goal is to set an agent's purpose, access, authority and limits up front, then let it work on its own within them. When it runs into something outside them, it can stop, ask for more context or hand the decision to a human expert.

Seen this way, more autonomy doesn't have to mean less control. **The control just moves.**

In a mostly human-run process, control usually comes from a person directing each step. In an agentic system, it increasingly comes from the structure around the agent: its context, permissions, tools and policies, how visible its actions are and the rules for when it should defer to a person. The work is in designing autonomy that earns trust.

Trust must be designed into the system



If more autonomy means control shifts away from directing every step, the next question is obvious: where does that control go?

It moves into the system around the AI.

That matters because trustworthy AI can't rest on the hope that a model is smart enough to make the right call every time. The safer path is to set up the conditions an AI works under before it ever takes action.

That starts with purpose. An agent needs a clear objective and a defined scope for what it's responsible for. The more specific that intent is, the easier it is to decide what information the agent needs, which tools it should use and where its authority should stop.

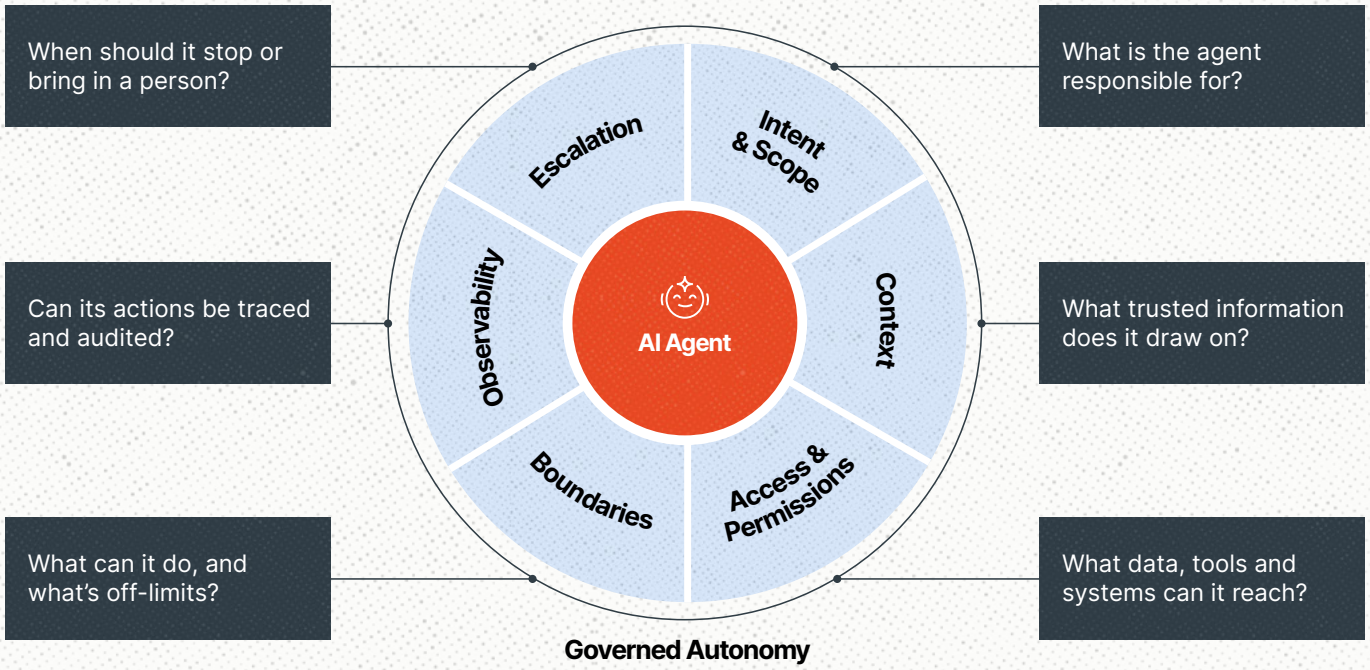
Context matters just as much. An AI system can only reason from what it knows, and general intelligence is a long way from understanding how a particular business actually works. An agent making calls about software quality might need requirements, process knowledge, test history or other trusted enterprise information.

Good context doesn't require unlimited access, though, and the same goes for data, tools and systems. An agent should be able to reach what it needs for the job, and by default nothing beyond that. What it can see, what it can call on and what it can change are three separate decisions.



Trust must be designed into the system (continued)

The System Around the AI



The intelligence sits at the center. The trust comes from everything around it.

As agents take on more responsibility, their actions need to leave a trail: what happened, which systems were involved and how it turned out. That trail lets teams investigate, validate and learn from what their agents are doing.

Then there are the boundaries. Sometimes the context is incomplete; the risk is higher than usual or the decision falls outside the agent's authority. Those moments need to be planned for when the system is designed, with clear conditions for when the agent keeps going, when it asks for more context and when the decision goes to a person.

The model brings the intelligence. The system around it sets the purpose, supplies the context, limits authority, controls access, records activity and decides when the AI should step back.

That's the shift that matters. You can't build trust on the assumption that the AI will always make the right decision. You build it by designing a system that gives AI room to act where it makes sense and makes clear where that ends.

UiPath AI Trust Layer: Putting guardrails around AI

Imagine an AI agent has been asked to help test a new customer payment workflow.

To do that well, it might need requirements, existing test assets, business context and testing tools. Some of that information could include sensitive customer data. The agent needs enough freedom to be useful, but unrestricted access would create exactly the risk enterprises are trying to avoid.

So the question shifts from "Do we trust the AI?" to something more practical: what should it be allowed to see, use and do, and how do we stay in control as its role grows?

That's where the UiPath Trust Layer comes in.

The AI Trust Layer governs how AI gets used. In practice, that means grounding AI in relevant enterprise context and protecting sensitive information with capabilities like PII masking. It means visibility into how AI is used and control over how much gets consumed. It also means choosing your own models through bring your own model (BYOM) and governing how AI connects to tools and systems through mechanisms like the Model Context Protocol (MCP).

So now, back to the payment-testing agent: it gets the business context it needs without being handed access to everything. Sensitive data stays protected, and the organization still controls its models, tool connections and usage.

The agent is every bit as capable as it was. What's changed is how carefully its environment has been put together. Good governance gives organizations the confidence to let AI do more without handing it unlimited freedom. The guardrails are what let it go further.

> **"You have autonomy, but you have it on your terms."**
>
> **Ingo Philipp,**
> **VP, Product Management, UiPath**



Delegation without giving up control

When you hand work to a colleague, you don't usually walk them through every step. You explain what you need, give them the context to do it well, set any important boundaries and make it clear when they should come back to you for a decision.

Consider this — a tester needs to validate a customer onboarding workflow. Instead of clicking through the application and spelling out every interaction, they can give **UiPath Delegate,** a multipurpose computer use agent, an instruction like: "Validate the customer onboarding process for a new account, including the required fields, verification steps and confirmation screen." The tester defines the outcome, not each step, and Delegate takes on more of the execution.

That's what separates delegation from telling AI to "do whatever it thinks is best." The person still sets the objective and the context. The AI has room to figure out how to get the job done, and that room sits inside the boundaries set for it.

The steps have changed hands. The terms of the work haven't.

The tester doesn't need to watch every interaction, and they don't drop out of the process either.

As this model matures, delegation moves people toward the parts of the work where their expertise counts most. AI handles more of the execution. People stay responsible for intent, judgment and anything that falls outside the expected path.



From human-in-the-loop to expertise-in-the-loop

"Human in the loop" has become familiar shorthand for responsible AI. It's a reassuring idea: if a person is involved, the AI can't really run on its own.

But the phrase raises a bigger question: which human?

> **"You want a human doing the oversight that has experience and that is engaged."**
>
> **Melinda Ballou,**
> **Research Director, AI Assurance, ALM, Quality and Portfolio Strategies at IDC Research**

Putting a person into a workflow doesn't automatically make it safer. If someone is asked to approve an AI decision without the context, expertise or authority to judge it, they can end up as little more than a rubber stamp. It looks like oversight, but no real judgment is happening.

A better model is expertise-in-the-loop.

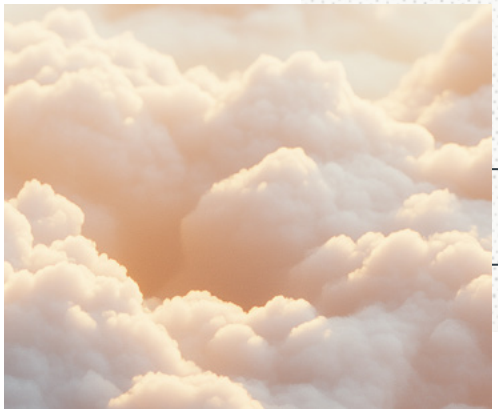
The idea is simple. AI works on its own when the work is well understood and inside its boundaries. When it hits ambiguity, higher risk or a decision that needs human judgment, it brings in the person best equipped to sort it out.

Who that is depends on the situation. A compliance question might go to a compliance expert. A conflict between business rules might need the process owner. A decision about test coverage, or about whether a result is an acceptable risk, might belong with an experienced tester.

That changes what the human is there for. Nobody needs to watch every move the AI makes. What matters is that when human judgment is needed, it comes from someone with the knowledge and authority to make the call.

This matters especially in software testing, which depends on understanding business intent, risk, expected behavior and what happens when something goes wrong. AI can handle more of the execution, but some answers will depend on context that's hard to fully capture. Our dark testing factory model follows the same thinking: people focus on intent, ambiguity, knowledge, exceptions and judgment, while agents and automations handle routine execution.

In the end, the loop is only as good as the people in it.



Humans should handle judgment and exceptions, not every action

When AI starts doing more on its own, the natural reaction is to put a person in front of every decision. But if someone must review every action an AI takes, most of the benefit of autonomy disappears.

Instead, let AI handle the work that's routine, repeatable and well understood, and to bring people in when a situation calls for judgment.

An agent might run a familiar testing workflow on its own, then ask for human input when it hits an ambiguous requirement, an unexpected result or a decision with real business or compliance implications. Human involvement can scale with risk and uncertainty.

Instead of watching automation run, people spend their time setting intent, defining policies, resolving ambiguity, handling exceptions and making the business calls AI can't responsibly make on its own.

It also avoids a common oversight trap: approval fatigue. When people are asked to sign off on too many routine decisions, approving turns into a reflex. Oversight means more when it's saved for the decisions that actually need expertise.

Those moments of human input don't have to be dead ends, either. A decision about an exception, a clarification of a business rule or a call on what the right outcome looks like can become reusable knowledge for the next run. Over time, each intervention can be worth more than the last.



Autonomy and control can coexist

It's easy to assume that more AI autonomy has to mean less human control. In practice, the two don't pull against each other.

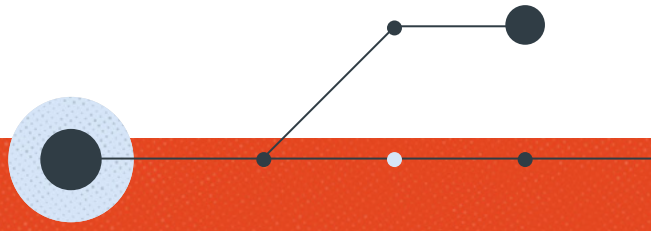
As AI takes on more of the execution, people move up, not out. AI brings the speed, scale and capacity to handle more of the work. AI brings the speed, scale and capacity to handle more of the work. People stay responsible for anything that takes context, expertise, judgment and accountability.

That's where governance comes in. Clear boundaries, controlled access, visibility into what agents do and defined escalation paths let AI work more independently while oversight stays in place. Test Cloud is built around this kind of control and traceability for agentic testing.

There's no need for a blanket rule that AI should always act on its own, or that a person should always be involved. The work is in deciding, deliberately, where AI can act independently and where human expertise adds the most.

> **"Accountability remains with the person no matter how much AI contributed."**
>
> **Melinda Ballou,**
> **Research Director, AI Assurance, ALM, Quality and Portfolio Strategies at IDC Research**

And that's what makes it possible to trust AI with more of the work.



About UiPath

UiPath develops AI technology that mirrors human intelligence with ever-increasing sophistication, transforming how businesses operate, innovate, and compete. The UiPath Platform™ accelerates the shift toward a new era of agentic automation—one where agents, robots, people, and models integrate seamlessly to drive autonomy and smarter decision-making. Powered by the UiPath Platform™, UiPath Test Cloud augments testers with enterprise-ready, production-grade, test automation for 190+ technologies and AI agents for agentic testing—the next evolution of software testing.

Want to see what governed autonomy looks like in practice?

Watch the webinar

UiPath