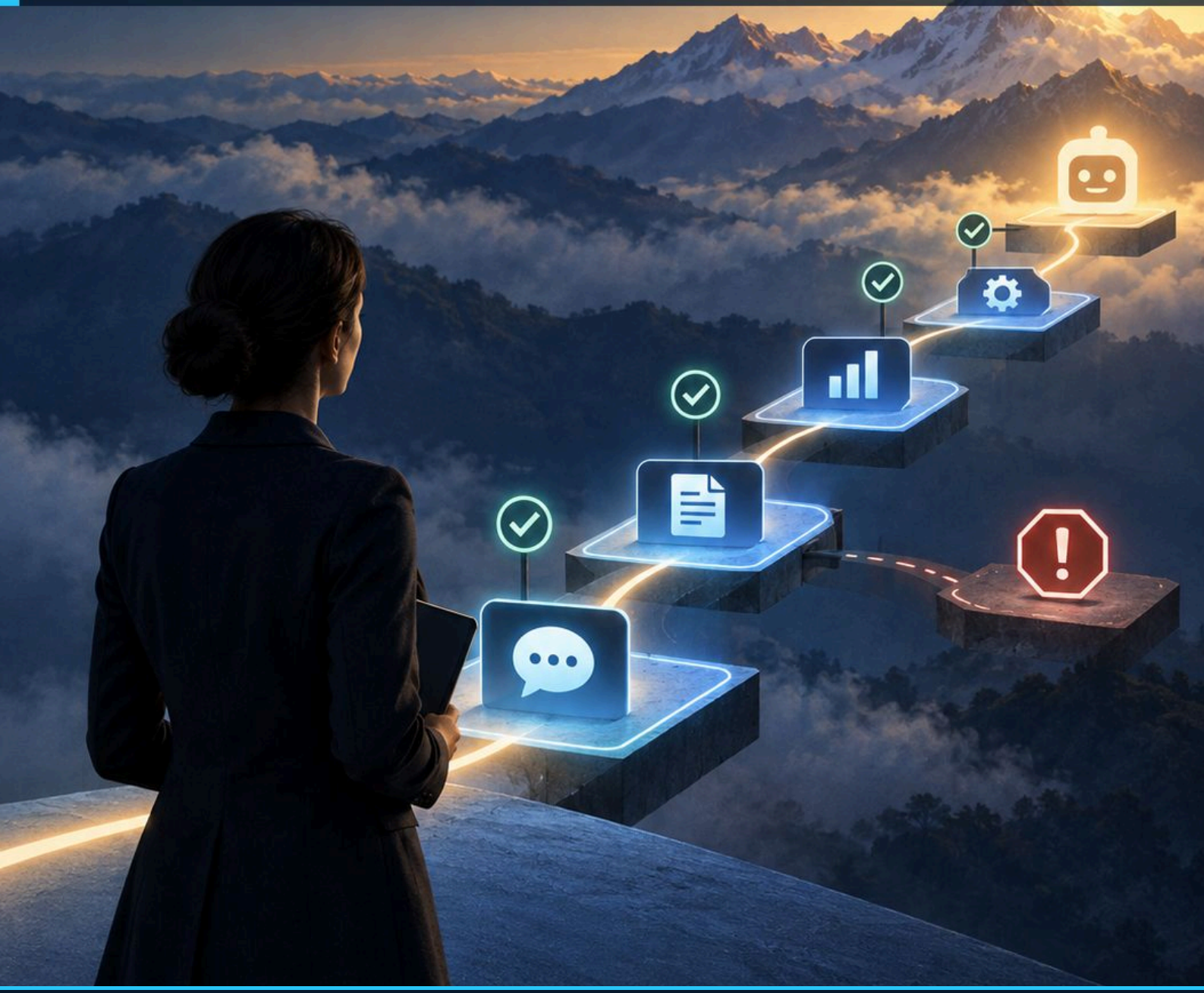
FROM ASSISTANTS TO EARLY AI AGENTS

An Enterprise Operating Model for Action-Taking AI

13 DECEMBER 2024 • ENTERPRISE AI SERIES



Paul Forrest

PAPER NINE · ENTERPRISE AI, 2023 TO 2024

About this paper

This is the ninth and final paper in a series on the enterprise adoption of generative artificial intelligence. Each paper is written to what was known when it was written.

The preceding eight concerned systems that produce content for a person to use. This one concerns systems that select tools and take actions, which moves the control question from reviewing output to governing permissions.

About half of what follows reports published evidence and vendor documentation. The rest is my own framework, which is a higher proportion than in any other paper in the series and reflects how little has been measured.

HOW TO CITE

Forrest, P. (2024). *From Assistants to Early AI Agents: An Enterprise Operating Model for Action-Taking AI.* Enterprise AI, 2023 to 2024, Paper Nine. Version 1.0, 13 December 2024.

HOW TO READ THE EVIDENCE TAGS

Every substantive claim carries a tag stating what kind of evidence stands behind it. A claim resting on two kinds carries both.

PREPRINT Research posted as a preprint or working paper, not peer-reviewed at this date.

VENDOR A vendor announcement or a figure the vendor reports about its own product.

AUTHOR My own framework or judgement, proposed and not yet proven.

A NOTE ON DATES

Research for this paper closed on 11 December 2024. Almost all the capability evidence comes from vendor publications of the last seven weeks, every figure is attributed to the vendor that produced it, and the paper says plainly where it is arguing.

Contents

ABSTRACT AND SUMMARY OF THE ARGUMENT

Systems that can act, evaluated honestly, which means cautiously

On 22 October 2024 Anthropic released a public beta allowing a model to use a computer, and described the capability in two documents published the same day.[1,2] VENDOR Its own research post states that computer use remains slow and often error-prone, and reports a score of 14.9 per cent on an evaluation of computer-using ability against a human range the post gives as 70 to 75 per cent.[2] VENDOR That is the best capability evidence there is at the date of this paper. It is also a vendor reporting on its own system.

The gap between that figure and the market conversation is the reason this paper exists. Systems that select tools and take multi-step actions are a genuine change in kind, because an error stops being a sentence a person can discard and becomes an action in a system of record. That change deserves an operating model now, while the capability is too weak to cause much harm and the habits are being set.

What follows defines an autonomy ladder with five rungs and excludes the sixth, sets out a threat model in which prompt injection is the distinctive entry, proposes a control architecture built on identity and least privilege, and gives a pilot gate. The organising claim is that as a system acquires agency, control shifts from reviewing content to governing permissions, actions and recoverability. AUTHOR

WHAT THIS PAPER ARGUES

When a system can act, the reviewable artefact disappears. Control has to move to what the system is permitted to touch, what it is permitted to do, and whether the action can be undone.

- The capability is real and weak. The vendor's research post reports 14.9 per cent on a computer-use evaluation against a stated human range of 70 to 75 per cent, its product post reports 22.0 per cent when the system is given more steps, and both figures are the vendor's own.[1,2] VENDOR
- The vendor's own framing is the most useful thing it published. The product post describes the capability as still experimental and imperfect and encourages developers to begin with low-risk tasks, and a supplier saying that about its own launch tells the reader more than any external assessment yet published.[1] VENDOR
- Prompt injection becomes a live control problem. Where a system reads screen content from the internet it may encounter content containing instructions, which the vendor names explicitly as a risk, and conventional input validation does not address it.[2] VENDOR
- Five rungs are in scope and the sixth is not. Answer, recommend, prepare an action, act with approval and act inside narrow bounds are discussed, while open-ended autonomy is excluded from this paper, because nothing in the published evidence supports it. AUTHOR
- Reversibility is the gate. A pilot on actions that cannot be undone is a deployment with a smaller user group. AUTHOR

SECTION 1

What an agent is, in December 2024

A working definition, chosen narrowly, because the word has been applied to everything from a scripted workflow to a research aspiration.

The term is being used this quarter for at least four different things. A chatbot with a persona, a deterministic workflow with a language model inside one step, a system that selects among tools to pursue a goal, and a speculative future system that operates without supervision. Only the third is a new control problem, and this paper uses the word only for that.

THE EARLY AGENT LOOP

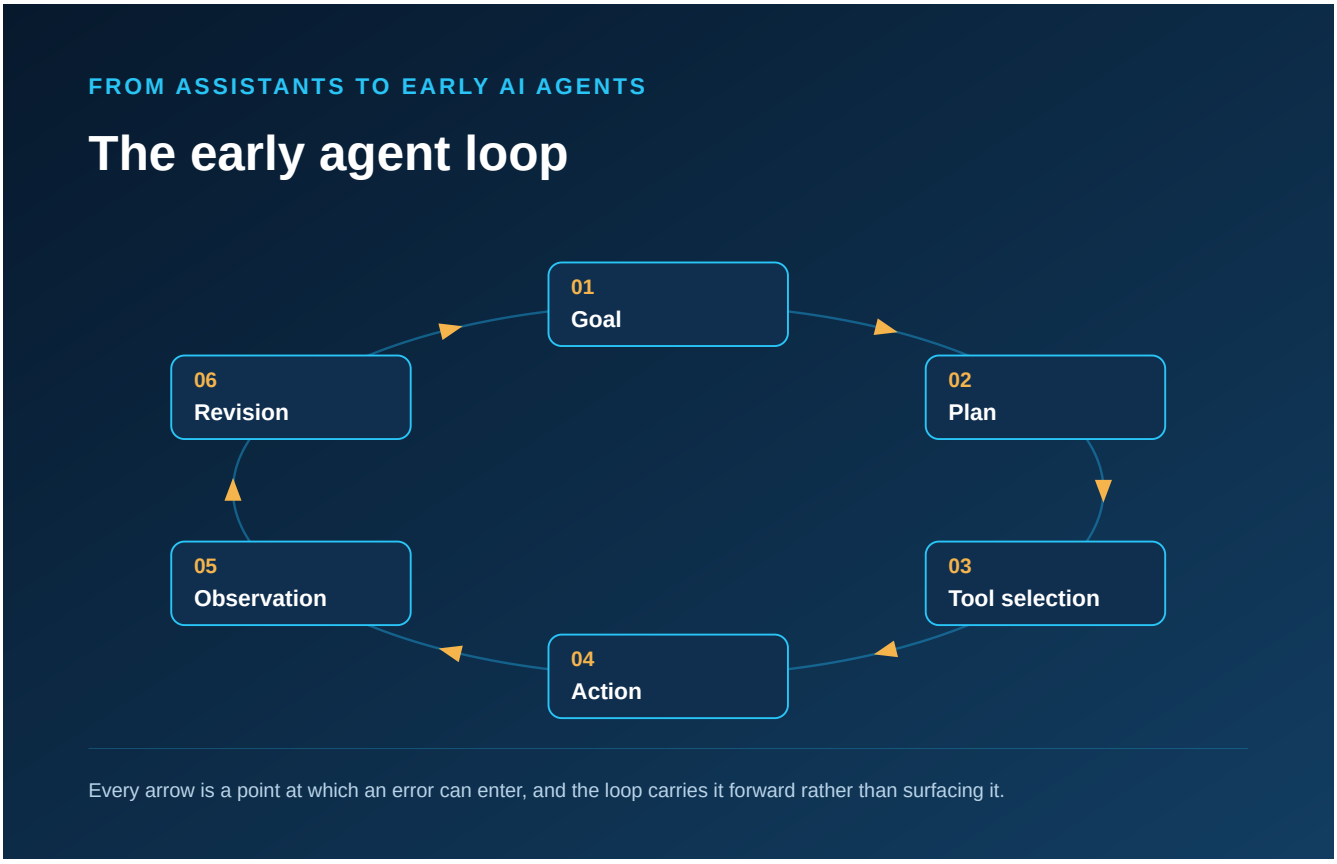


FIGURE 1 Goal, plan, tool selection, action, observation and revision. Every arrow is a point at which an error can enter, and the loop is what distinguishes these systems from a sequence of prompts.

My framework. AUTHOR

SPECIFICATION

The working definition used throughout this paper

- The system pursues a goal expressed by a person rather than executing a specified sequence.
- It selects among available tools or actions, and the selection is made by the model rather than by a rule the organisation wrote.
- It operates over multiple steps, using the result of one step to decide the next.
- It has some effect beyond producing text, whether that is writing to a system, sending something, or changing a state.

The distinction that matters for governance is between deterministic automation and model-directed action. An organisation that has automated a process with rules knows what the system will do, because somebody wrote it down. An organisation that has given a model a goal and a set of tools knows what the system is permitted to do and cannot know in advance what it will choose. The controls in this paper all follow from that difference. AUTHOR

Why the loop is the thing

A sequence of prompts is not an agent, however long the sequence. A person decides each time whether to continue. The loop closes when the system observes the result of its own action and decides the next step without a person in between. That closure is what produces the useful behaviour and it is also what produces cascading error, and the two cannot be separated by design.

SECTION 2

What the capability evidence actually shows

One vendor beta, two documents, and a set of figures the vendor reports about its own system. Read carefully, they counsel caution.

On 22 October 2024 Anthropic announced a public beta enabling a model to use a computer in the way a person does, publishing a product announcement and a research post on the same day.[1,2] VENDOR These are the primary capability evidence available at the date of this paper, and both are the vendor's own account.

The product announcement describes the capability as still experimental, says that the model's current ability to use computers is imperfect, and encourages developers to begin exploration with low-risk tasks.[1] VENDOR The research post says that computer use remains slow and often error-prone.[2] VENDOR It is unusual for a launch to be framed this way and the framing should be taken at face value.

The numbers, and which document each comes from

1. The research post reports a score of 14.9 per cent on an evaluation of computer-using ability, against human performance the post gives as generally 70 to 75 per cent, and notes that the next best system scores 7.7 per cent.[2] VENDOR
2. The product announcement reports the same 14.9 per cent in a screenshot-only category against a next-best figure of 7.8 per cent, and adds that when afforded more steps to complete the task the system scored 22.0 per cent.[1] VENDOR
3. The evaluation itself is an independent benchmark described in an April 2024 paper, covering several hundred real-world computer tasks in executable environments.[3] PREPRINT

The honest summary is the best available system at roughly a fifth of human performance. A system completing between 15 and 22 per cent of tasks depending on conditions, against people completing around three quarters, is a genuine research result and a poor basis for unattended enterprise work. The correct conclusion is that this is the moment to build the operating model, not the moment to deploy.

Two cautions belong with these figures. They are vendor-reported evaluations of the vendor's own system, which is the normal state of affairs in this field and remains worth stating. And a benchmark measuring computer-using ability across general tasks tells an organisation little about performance on its own narrow task, which is the only number that would justify a deployment. AUTHOR

The connective layer

On 25 November 2024 Anthropic published an open standard for connecting assistants to the systems where data lives, released with specifications, software development kits, local server support in its desktop applications and a repository of pre-built connectors.[4] VENDOR Several early adopters were named in the announcement.[4] VENDOR

Standardised connection matters more for governance than for capability, because it lowers the effort required to give a system reach into enterprise data. A control regime that depended on integration being difficult will stop working. At the date of this paper it is one vendor's published standard with a handful of named adopters, and nothing supports describing it as an industry standard. AUTHOR

SECTION 3

An autonomy ladder with a deliberate ceiling

The ladder has five rungs, and the step from preparing an action to taking one is where the control problem changes character.

Autonomy is treated in most current discussion as a single dimension running from helpful to dangerous. It is more useful as a set of discrete rungs, because each rung has a different control requirement and organisations can occupy one deliberately for a long time.

THE AUTONOMY LADDER

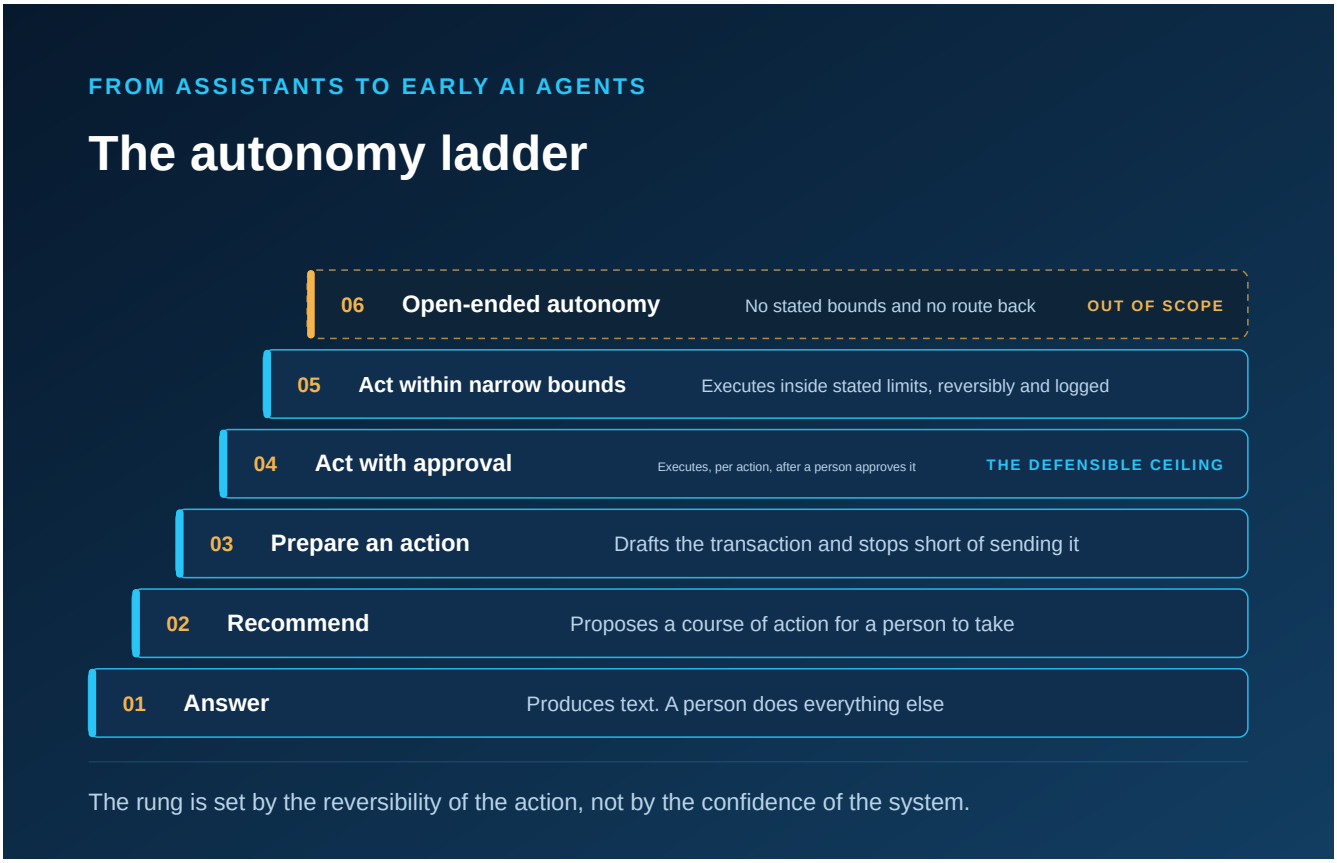


FIGURE 2 Five rungs, each defined by what the system may do without a person. The rung is set by the reversibility of the action rather than by the capability of the model, and the step from preparing to acting changes the control problem.

My framework. AUTHOR

FIVE RUNGS, AND WHAT EACH REQUIRES

RUNG	WHAT THE SYSTEM DOES	WHAT HAS TO BE TRUE BEFORE THIS RUNG IS USED
Answer	Produces information for a person. No tools, no actions, no external effect.	The controls from the earlier papers in this series. Verification rung named, data boundary observed.
Recommend	Proposes a course of action with reasoning, which a person then takes or declines.	The person can see what the recommendation rests on, and has the standing and time to decline it.
Prepare an action	Assembles a draft transaction, message or change, which a person reviews and executes.	The prepared item is legible in full before execution. Preparing something a reviewer cannot fully inspect is acting by another route.
Act with approval	Executes on confirmation, one action at a time, within an allow-listed set.	Approval is meaningful rather than habitual, actions are individually approved, and the approver can see what will change.
Act within narrow bounds	Executes without per-action approval inside a defined and limited envelope.	Every action reversible, hard transaction limits, complete logging, monitoring with alerting, a tested stop, and a named owner watching.
Open-ended autonomy	Out of scope for this paper.	Nothing in the published evidence supports it in an enterprise setting at this date, and no control set proposed here would make it safe.

The third rung is not as safe as it looks

Preparing an action for human execution sounds like a control and frequently is not. Where the prepared item is a 100-line configuration change, a bulk update or a message to a list, the reviewer cannot meaningfully inspect it and will approve it on the basis that it looks right. That is the fourth rung dressed up as the third. The test is whether a reviewer can fully inspect what they are approving in the time available. AUTHOR

Why the sixth rung is excluded rather than discussed

Open-ended autonomy appears in a great deal of current writing, generally as a destination. This paper leaves it out. The published capability evidence sits at around a fifth of human performance on general computer tasks, no evaluation method exists for unattended enterprise operation, and no control set proposed here would make it defensible. Treating it as out of scope is more honest than listing controls for something that should not be attempted.

SECTION 4

A threat model with one unfamiliar entry

Six of the seven failure modes are variants of problems security functions already know. The one that leads the table is new, and it is the likeliest to be mishandled.

Most of what can go wrong with an action-taking system is recognisable. Over-privileged accounts, compounding errors, data leaving the boundary and unreconstructable incidents are all familiar, and existing disciplines apply with modest adaptation.

SEVEN FAILURE MODES, AND WHAT ADDRESSES EACH

FAILURE MODE	WHAT IT LOOKS LIKE	THE CONTROL THAT ACTUALLY ADDRESSES IT
Prompt injection	Content the system reads contains instructions it follows, from a web page, a document or a message.	Treat all read content as hostile. Separate systems that read untrusted content from systems that can act. Input filtering does not solve this, because instruction and data are the same thing.
Wrong tool selection	The system chooses a plausible but incorrect action, and executes it competently.	A short allow-list per task, with tools the system may not reach absent from the environment rather than merely forbidden.
Privilege escalation	The system reaches data or functions beyond the task, usually through an over-provisioned service account.	A distinct identity per agent with least privilege, scoped to the task and time-bounded. Never a shared or human credential.
Cascading error	An early mistake becomes the input to later steps and compounds invisibly.	Step limits, checkpoints where state is verified, and a budget of actions after which the loop halts for review.
Data exfiltration	The system moves material outside the boundary, often as an unremarkable step in a plausible plan.	Egress controls at the network and tool layer, not in the prompt. Instructions to the model are not a security boundary.
Irreversibility	An action cannot be undone, and the cost lands outside the organisation.	Reversibility as a gate on the rung. Where an action cannot be undone, it does not belong above the third rung.
Weak auditability	Something happened and nobody can reconstruct the sequence that produced it.	Log the plan, the tool calls, the parameters, the observations and the outcome. A log of outputs alone is insufficient once the system acts.

Prompt injection, and why the usual instincts fail

Where a system reads content it did not originate, that content may contain instructions the system follows. The vendor names this directly, noting that because the model can interpret screenshots from computers connected to the internet it may be exposed to content that includes prompt injection attacks.[2] VENDOR

Two conventional responses do not work. Input validation fails because the instruction and the data are the same text, and no filter can reliably distinguish content that describes an instruction from content that is one. Instructing the model to ignore instructions in the content fails because that instruction is itself in the same channel and carries no privileged status. The architectural answer is separation, keeping systems that read untrusted content away from systems that can act, and where the two must meet, requiring approval at the boundary. AUTHOR

A NOTE ON THE VOLUNTARY RISK GUIDANCE

The generative profile published in July 2024 addresses several of these failure modes and is a reasonable source for suggested actions. It is voluntary, it predates the public availability of computer-using systems by three months, and it was not written with action-taking in view. Use it for the risks it covers and no further.

SECTION 5

A control architecture built on permissions

Nine controls follow, and reviewing what the system produced moves from the centre of them to the edge.

The controls in the earlier papers assume a reviewable artefact. A person reads the output and decides whether to use it, and the verification ladder from the second paper sets how hard they look. Once the system acts, that artefact may not exist and the action may already have happened by the time anybody reads anything.

CONTROLS FOR ACTION-TAKING SYSTEMS

FIGURE 3 Nine controls, of which identity and least privilege carry the most weight. Content review sits deliberately at the edge, because reviewing what the system said stops being the primary control once it can act.

My framework. AUTHOR

1. A distinct identity, so that each agent has its own credential, never a shared service account and never a human's. Without this, nothing in the log can be attributed.
2. Least privilege, time-bounded, with permissions scoped to the task and expiring with it. An agent provisioned with standing broad access is the commonest design error in early deployments.

3. An allow-list of tools, short, explicit, and enforced by the environment. Tools absent from the environment beat tools the system has been told not to use.
4. A sandbox, meaning an execution environment where actions are contained and observable, used for everything until the pilot gate is passed.
5. Approval gates at the fourth rung, per action, showing the approver what will change.
6. Limits on transaction value, action count, rate and time window, enforced outside the model, because a limit the model is told about is a suggestion.
7. Observability, retaining the plan, the tool calls with parameters, the observations and the outcomes long enough to investigate.
8. Evaluation against a task set specific to the deployment, run before release and after any model change, measuring completion and the frequency of wrong actions as well as output quality.
9. A stop, meaning a mechanism to halt a running system immediately, tested on a schedule, with a named person authorised to use it without convening anybody.

The control that is usually theoretical

Almost every deployment claims a stop mechanism and few have tested one. The test is worth describing because it is cheap. Have somebody who is not the developer stop the system mid-task, at a time nobody arranged, and record how long it took, what state the work was left in, and whether anybody downstream noticed. An untested stop is an assumption, and it is the assumption everything else rests on. AUTHOR

SECTION 6

Who owns what

Five parties own the pieces. The pattern follows the sixth paper, with one addition that changes the security function's workload.

The operating model for action-taking systems is not a new structure. It is the structure from the sixth paper of this series with a heavier security involvement and one boundary drawn more firmly, which is the boundary between the people who build these systems and the people who may authorise them to act.

1. The artificial intelligence office owns portfolio coherence, the autonomy rung assigned to each deployment, the evaluation standard and the reporting.
2. The technology function owns the platform, the sandbox, the identity arrangements and the tool allow-lists as an operated service.
3. The security function owns the threat model, the injection testing, egress control and the incident route. This is materially more work than for content-generating systems and should be resourced as such.
4. The business process owner owns the task definition, the reversibility assessment, the approval design and the consequences of the actions taken.
5. Independent assurance tests whether the described controls are the operating controls, with particular attention to whether the stop has ever been exercised.

The boundary to hold

The team that builds an agent should not authorise it to act in production. This is ordinary segregation of duties and it is under more pressure here than usual, because the people who understand these systems are few and are concentrated in the building team. Where an organisation cannot separate the two, the honest position is that it is at the third rung and should stay there until it can. AUTHOR

SECTION 7

A pilot gate, and what to do in the first quarter

Seven requirements follow, each with a test, and an organisation that cannot pass them is not ready, which at this date is the expected finding.

The recommendation in this paper is narrow. Build the operating model now, run one bounded pilot in a sandbox, and do not put an action-taking system into production against consequential work this quarter. The capability evidence does not support it and the vendors say so themselves.

THE PILOT GATE, WITH A TEST FOR EACH ITEM

REQUIREMENT	WHAT IT MEANS	HOW TO TEST THAT IT IS TRUE
Bounded task	One task class, described narrowly enough that the boundary is checkable.	Two people independently classify 20 real cases as in or out of scope, and agree.
Reversible actions	Every action the system may take can be undone within the pilot.	Undo each action type once, in the test environment, and time it.
Test environment	A representative environment that is not production.	The system runs the full task end to end there before any production access is discussed.
Measurable success	A definition of a completed task and a quality standard, written first.	Somebody who was not involved can apply the definition to an output and reach the same verdict.
Red-team cases	Deliberate attempts to induce the wrong action, including injected instructions in content the system reads.	At least one injection attempt succeeds during testing. If none does, the testing was not adversarial enough.
Fallback	What happens when the system stops, fails or is stopped.	Turn it off mid-task and confirm the work can be completed by a person without data loss.
Named incident owner	A person, reachable, with authority to stop the system.	A named owner who cannot be reached out of hours is a documented intention, so telephone them and find out.

The red-team requirement, stated as a pass condition

Most adversarial testing in this area is performed by the team that built the system and finds what that team expected to find. The condition above is deliberately awkward. At least one injection attempt should succeed during testing, because a test programme that produces no successful attacks has demonstrated the limits of its own imagination. Where nothing succeeds, bring somebody in who did not build it. AUTHOR

WHAT TO DO BETWEEN NOW AND THE SPRING

Stand up a sandbox and put one bounded, reversible task through it end to end. Write the autonomy rung definitions and get them agreed by the security function and the business owner. Test a stop mechanism and record the result. Extend the inventory from the eighth paper to record which systems can act and what they may touch.

That is a quarter of work, it produces no deployment, and it is the difference between having an operating model when the capability becomes usable and building one under pressure afterwards.

A closing note on the series

Nine papers over nearly two years have described a technology moving faster than the methods for evaluating it, and the argument has been the same one throughout. What an organisation can responsibly claim is set by what it can evaluate. That capability has been the binding constraint at every stage. That has not changed as the systems became more capable, and on the evidence in this paper it has become more acute, because the artefact that made evaluation possible is the thing action-taking removes.

Whether the figures in section two improve quickly is not knowable here. Seven weeks of public beta establishes no trend, and extrapolating from seven weeks is how most of the confident predictions in this field have been produced. The organisations that will be in the best position when the answer arrives are the ones that spent this quarter building the mechanism.

END MATTER

Method, evidence and limits

How the sources in this paper were chosen and dated, and what the reader should distrust. This edition requires more of that warning than the previous eight.

Where the evidence comes from

The capability evidence is vendor documentation, because at this date no independent evaluation of enterprise action-taking systems exists. Every figure is attributed to the vendor in the sentence that uses it, and the two documents published on 22 October are cited separately because they report different figures and different framings.

On citing two posts from the same day

The product announcement and the research post give different comparator figures and only the research post states the human range. Merging them produces a quotation that appears in neither. I have kept them apart.

The proportion that is my own

Roughly half of this paper is my own framework, which is a higher proportion than any other paper in the series. That reflects the state of the evidence. Where almost nothing has been measured, a paper that confined itself to reporting would be two pages long and would not help anybody.

On the benchmark paper

The evaluation environment described in section two is cited from its April preprint. The paper was accepted for a conference that meets this month, and the proceedings version had not appeared when research closed. It carries the preprint tag for that reason. The figures attached to it come from the vendor, and the text says so wherever they appear.

What does not exist yet

No independent measurement of action-taking systems in enterprise settings. No evaluation method for multi-step agent behaviour that an organisation could adopt. No published defence against prompt injection that is understood to be reliable, which is the gap in this paper I would most like to close. And no evidence on how quickly the capability figures will move.

Everything in this paper that is mine

The working definition, the five-rung ladder, the threat model, the nine controls, the operating model and the pilot gate are all my own, and none has been tested at scale, because there is no scale at which to test it yet. The argument that the third rung is frequently the fourth in disguise is a practitioner observation.

Interest declared

I advise on technology adoption and the operating model work described here is work I could be engaged to do. The paper recommends against production deployment this quarter, which is the larger engagement, so the recommendation runs against my interest.

END MATTER

Sources considered and set aside

Announcements and benchmarks I read and set aside. In a field this young most of what circulates is a press release.

SOURCE	PUBLISHED	WHY IT WAS SET ASIDE
Gartner, top strategic technology trends for 2025	21 October 2024	Names agentic artificial intelligence as the first trend of next year and forecasts adoption. It is a forecast, from a firm whose method is not published.
OpenAI, SWE-bench Verified	13 August 2024	A coding benchmark, human-validated, and confined to software repair. Relevant to one narrow class of action and not to the general case this paper is about.
Zhou and others, WebArena	25 July 2023	An academic benchmark for browsing agents, over a year old and confined to web tasks. The April 2024 environment cited in section two supersedes it for this purpose.
Cognition, Devin announcement	12 March 2024	A vendor demonstration of a software agent with one benchmark figure and no public access at this date. Nothing in it can be checked.

END MATTER

About the author

Paul Forrest advises boards and executive teams on the adoption of new technology in operating businesses, with a practice grounded in delivery rather than in advisory work alone.

His background is in business process reengineering, which he has practised since the 1990s, and in applied natural language processing, which he has worked on since 2014. The insistence on reversibility in this paper comes from the first of those. Process automation programmes taught, expensively, that the question worth asking about any automated step is not how often it is right but what it costs to undo when it is wrong.

He writes on the condition that a claim carries its evidence, and this series has been an attempt to hold that line while a subject moved faster than the evidence about it.

END MATTER

References

Four works cited in the text, each with its original date of publication. None is later than 11 December 2024.

1. **Anthropic.** *Introducing computer use, a new Claude 3.5 Sonnet, and Claude 3.5 Haiku.* Product announcement. Cited for the description of the capability as still experimental and imperfect, for the encouragement to begin with low-risk tasks, for 14.9 per cent in the screenshot-only category against a next-best figure of 7.8 per cent, and for 22.0 per cent when afforded more steps.
   PUBLISHED 22 OCTOBER 2024

2. **Anthropic.** *Developing a computer use model.* Research post published the same day. Cited for the statement that computer use remains slow and often error-prone, for 14.9 per cent against a stated human range of 70 to 75 per cent and a next-best figure of 7.7 per cent, and for the statement on exposure to prompt injection through interpreted screen content.
   PUBLISHED 22 OCTOBER 2024

3. **Xie, T., Zhang, D., Chen, J., and others.** *OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments.* arXiv:2404.07972. Several hundred real-world computer tasks in executable environments.
   PUBLISHED 11 APRIL 2024

4. **Anthropic.** *Introducing the Model Context Protocol.* Cited for the described scope of the standard, the three components of the release and the named early adopters, all attributed to the vendor.
   PUBLISHED 25 NOVEMBER 2024

END MATTER

Further sources consulted

Seven sources I read for this paper and did not cite. They are listed so that the reading behind it can be seen in full, and none is later than 11 December 2024.

- **National Institute of Standards and Technology.** *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile.* NIST AI 600-1. Voluntary, and published three months before computer-using systems became publicly available.
  PUBLISHED 26 JULY 2024
- **European Union.** Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence. In force from 1 August 2024, with substantive obligations applying from 2 February 2025 onwards.
  PUBLISHED 12 JULY 2024
- **Microsoft.** *Introducing Copilot Actions, new agents, and tools to empower IT teams.* Cited as a vendor announcement.
  PUBLISHED 19 NOVEMBER 2024
- **Salesforce.** General availability of its agent platform, following its unveiling on 12 September 2024. Cited as a vendor announcement.
  PUBLISHED 29 OCTOBER 2024
- **OpenAI.** Release of the full reasoning model and a new subscription tier.
  PUBLISHED 5 DECEMBER 2024
- **National Institute of Standards and Technology.** *Artificial Intelligence Risk Management Framework 1.0.* NIST AI 100-1.
  PUBLISHED 26 JANUARY 2023
- **United States Office of Management and Budget.** *M-24-10.* Cited for the decision rights pattern discussed in section six.
  PUBLISHED 28 MARCH 2024

THE END OF THE SERIES

Nine papers, one argument

The series began in January 2023 with a conversational system that had been public for two months and an argument that the useful first capability was a learning system. It ends with systems that can act, and the same argument in a more urgent form.

Every paper was written to the evidence available when it was written. Where a later paper contradicts an earlier one, the earlier one stands as published, because a record of what was reasonable to believe at the time is worth more than a document quietly corrected.

WHAT THE NINE PAPERS ASKED AN ORGANISATION TO BUILD

- An evaluation capability, which was the binding constraint in January 2023 and remains it.
- A record of what exists, what it does, what evidence supports it and who decided.
- A named person who may accept residual risk, and one who may stop a running system.
- The judgement to tell when the output is wrong, which no product release has yet removed the need for.

Paul Forrest · 13 December 2024 · Version 1.0 · Research closed 11 December 2024