

Cheap to attempt, expensive to trust.

OpenAI shipped a model that can operate software for forty minutes at a stretch and still fails most of the way through six out of ten professional automations. Seven businesses that work anyway, because in each one checking the answer costs less than producing it. Each also gets the specific reason it might not.

BY DAVE BROWN

GPT-6 Astra shipped on September 3rd and the argument started before the docs finished loading. OpenAI's president said he personally thinks we're there. The company's own launch materials pointedly did not say that, settling for "most intelligent and aligned model" and "world's best computer use model." Everyone else has spent the week relitigating what AGI means.

That argument is unfalsifiable and it is not the one that pays. There is a narrower question sitting underneath it with an actual answer: **what work is now worth doing that wasn't worth doing last month?** Answering it requires no position on machine consciousness, only two numbers: what a unit of work costs, and what it costs to check it.

Every company has work worth doing that nobody does, because sending a human to go and get it costs more than the work returns.

The honest read of the launch is that this is not a general leap. Independent evaluation puts Astra roughly level with its own predecessor on broad intelligence and behind Anthropic's Fable 5.1. What moved, and moved hard, is the ability to sit in front of bad software and keep going: longer horizons, fewer wasted tokens, a hallucination rate that fell by roughly half, and task times that dropped by nearly half as well.

That is a strange, specific capability, and it favors a strange, specific class of business. What follows is seven of them, with the same discipline as everything else I write: every idea gets a matching paragraph on how it dies, and most of them have a funded competitor already.

41.4%

AutomationBench. The success rate every business here has to survive

37 pts

gap between the headline ARC-AGI-3 score and the independent harness's

2.5×

the direction frontier token prices moved this month. Up.

7

businesses here, each with the reason it might not work

The seductive version of this thesis is "intelligence too cheap to meter, spin up a thousand interns, print money." Astra costs two and a half times what the model it replaced cost, per token, and the independent measurement of cost per completed task came back *higher*. Anyone budgeting off the long-run price curve should check how far away the long run is.

—— WHAT'S IN HERE

- 01 What actually shipped. The capability, separated from the launch narrative, with confidence levels attached.
 - 02 The filter. Why the cost of checking the work decides everything downstream of it.
 - 03 Exhaustive search. Three businesses that read everything nobody could afford to read.
 - 04 Operating bad software. Three aimed at the interfaces that will never have an API.
 - 05 The strange one. A service company with no employees and no dashboard.
 - 06 What has to be true. Five conditions, and the one objection I have no answer to.
 - 07 Signposts and a cheap test. What to watch, and thirty days that beat a market map.
-

A specific capability improved. Most of the coverage described a different one.

Worth pulling the numbers apart before building on any of them. The headline figures and the useful figures turn out to be different sets, and at least one headline does not survive contact with an independent harness.

Every row below carries a confidence level, since that is the column that changes what you do about it.

THE CLAIM	THE NUMBER	HOW SOLID
Computer use	72.6% on OSWorld 2.0 against 65.7% for GPT-5.6 Sol, and roughly 40 minutes per task where Sol took about 75.	OpenAI-reported, on an offline subset. The time figure matters more than the score.
Professional automation	41.4% on AutomationBench, against 31.4% for Fable 5.1 and 18.1% for Sol.	A large relative jump. Also: still fails most tasks. Both facts are load-bearing.
Long-horizon work	59.3% on Agents' Last Exam, ahead of Opus 5 at 55.5% and Sol at 53.6%, using far fewer output tokens.	Leads the public leaderboard. The benchmark is still filling out toward its target task count.
"Saturated" ARC-AGI-3	99.9% under OpenAI's Provider Adapter harness. 62.7% under ARC Prize's standard harness.	The most important caveat of the launch. A 37-point spread living in the scaffolding rather than the weights.
General intelligence	Artificial Analysis Intelligence Index: 61.2, against 60.9 for its own predecessor and roughly 65.7 for Fable 5.1.	Essentially flat. Whatever happened here, it was not a broad IQ increase.
What got worse	Roughly 80 Elo points lost on GDPval-AA v2, plus small regressions on banking tool use, scientific coding, and long-context reasoning.	Real, and rarely mentioned. Partly an artifact of benchmark shape, since GDPval is one-shot, though not entirely.

THE CLAIM	THE NUMBER	HOW SOLID
Hallucination	AA-Omniscience hallucination rate roughly halved, from 92% to 51%.	The most underrated number in the launch, for reasons Part two is entirely about.
Price	\$10 per million input tokens, \$50 output. Two and a half times the model it replaces.	Solid, since it is a price list. Fast mode doubles it again.
Access	Staged rollout, vetted organizations first, message caps on consumer tiers, and a Critical cybersecurity classification restricting some capability.	Plan for capacity you don't control. This is not a commodity you can just buy in volume today.

Two rows deserve to come out of the table, because people are getting both of them wrong in opposite directions.

- **Read the harness gap as a lesson rather than a scandal.** A 37-point swing from changing how reasoning state and context are managed tells you that a large fraction of usable capability sits in the scaffolding around the model rather than inside it. The same trick previously moved the prior model from 13.3% to 38.3%. For anyone building here that is good news, because scaffolding is the part you control, and it is why “wait for the next model” is a weaker strategy than it sounds.
- **Nobody has a reliable cost model yet.** One widely circulated estimate had per-task cost falling roughly 57% on computer-use work because Astra burns fewer tokens getting there. An independent evaluator measured cost per task across a broader mix and found it about 75% *higher*, because the price increase outweighed the efficiency gain. Both are probably right about their own task mix. That they are both circulating means your unit economics are something you will have to measure for yourself.

THE ONE FACT THAT MATTERS MOST

The hallucination rate fell by roughly half while the automation success rate sits at 41%. Those two numbers together describe a worker who is wrong often but increasingly willing to tell you so. That combination is worth far more commercially than a smarter worker who is confidently wrong, because it makes the checking step cheap. The checking step is where every business in this document actually lives.

Whoever can check the work cheaply owns the business.

One test separates the businesses that work at a 41% success rate from the ones that need a capability nobody has shipped, and agent intelligence is not the variable in it.

A 41% success rate sounds like a reason not to build. It isn't, provided you never sell the agent's raw output. You sell *verified* output, and the economics of the whole business reduce to one ratio: what it costs to attempt the work, divided by what it costs to know whether the attempt succeeded.

Checking is nearly free in some workflows. A test suite passes. A checksum matches. An invoice either duplicates an earlier one or it does not. A form submits and returns a confirmation number. Where that holds, a 41% success rate becomes a scheduling problem: run it three times, keep whatever verifies, and delivered accuracy lands near 80% at three times the compute cost, which is still a rounding error against the human alternative. Where checking instead requires a licensed professional to read the output carefully, you have not built a business. You have built a way to generate work for a licensed professional.

Run the same task three times, keep whatever verifies, and a model that fails six times in ten delivers work that is right four times in five.

This is also why the long-horizon research matters more than the launch benchmarks. Errors in agent workflows are positively correlated across steps rather than independent, so failure probability grows faster than the naive compounding math predicts. Current frontier agents are close to perfect on tasks a skilled human would finish in under four minutes and succeed less than a tenth of the time on tasks that would take a human more than four hours. Success rates of 40–50% on short tasks fall below 10% when those same tasks are embedded in a long interaction history.

The design implication is unambiguous and almost nobody's pitch deck reflects it: **decompose until every unit fits inside the reliable horizon, then verify at every seam**. Sixty independently checkable four-minute units will finish work that one four-hour autonomous run will not.

THE FILTER, STATED ONCE

High-value work × enormous volume of tedium × objectively checkable output × a buyer already paying humans to do it badly. All four have to be present. Three out of four has been the shape of most AI pilots that quietly ended.

Everything in Parts three through five passes that filter. Whether each one passes the harder tests in Part six is a different question, and for at least two of them the answer is probably no.

PART THREE · EXHAUSTIVE SEARCH

Three businesses built on reading everything.

These share one mechanism. There is a pile of documents nobody reads all of, because reading all of it costs more than the value hiding in it. Move the cost of reading and the arithmetic inverts.

Most organizations leave money on the floor in amounts too small to justify a human going to get it. Nobody pays an analyst \$300 to chase a \$137 duplicate charge. The entire market here is that threshold, the point below which recovery costs more than it returns, and it just moved.

IDEA 01

The recovery agent.

Connect the books, the contracts, the CRM, and the inbox. Then run dozens of independent searches against them, continuously. Duplicate payments. Invoices that don't match the contract they were issued under. Credits never claimed. Subscriptions nobody has opened in a year. Rate increases that slipped through unchallenged. On the revenue side, the mirror image: quotes never followed up, inbound leads never answered, completed work never invoiced, customers who reliably reorder every six months and haven't.

Price it on contingency and the token-cost problem solves itself structurally, since compute only gets spent on volume you are already taking a percentage of, and "no savings, no bill" removes most of the sales friction. The pitch to a contractor never mentions architecture. It is *we found \$14,860 of work your office forgot about last month.*

WHO BUYS

Mid-market operators and trades businesses with messy back offices

HOW IT PRICES

20–30% of recovered dollars, the established rate in this category

VERIFICATION COST

Near zero. A duplicate is a duplicate; the recovered dollar is the proof.

THE RISK

Recovery audit is a billion-dollar existing industry with entrenched firms, and they are not asleep. They hold the ERP integrations, the vendor relationships, the audit rights written into supplier contracts, and the muscle memory for actually collecting on a finding, which is the hard half. Large enterprises have already been swept repeatedly; the unclaimed pool sits in businesses too small for the incumbents to bother with, where the dollars per account are correspondingly small. AI makes finding the money cheap and does nothing about getting a vendor to send it back, which stays a slow, relationship-bound collections problem.

IDEA 02

Exhaustive sourcing.

The instruction is one sentence: *get me 800 feet of this, delivered Thursday, at the lowest landed cost*. Then you run it wide. Distributors, manufacturers, regional suppliers, equivalents and substitutes, stock levels, lead times, freight math, and the quotes themselves. A purchasing manager under time pressure calls six vendors. The economically rational number to call was always closer to six hundred, and nobody has ever been able to afford it.

This is the cleanest example in the document of a general pattern worth naming: **exhaustive search becomes economically viable**, and that hands small companies a procurement posture that previously required a purchasing department. Construction materials, electrical parts, industrial components, and MRO supplies are where the spreads are widest and the catalogs worst.

WHO BUYS

Contractors, small manufacturers, anyone buying physical inputs without a purchasing team

HOW IT PRICES

Percentage of documented savings, or subscription once the baseline is trusted

VERIFICATION COST

Low, though savings get measured against a counterfactual you chose

THE RISK

The other side of this transaction is a human being who does not want to answer six hundred automated quote requests. The moment this works at scale, suppliers put up gates, quote portals add friction, sales teams start ignoring anything that smells automated, and the exhaustive search you sold gets throttled into an ordinary one. It is a defection strategy that stops paying once everybody defects, which is the same arc emailed outreach and cold LinkedIn both ran. There is also a measurement problem hiding in the pricing model: "savings" is the gap between what they paid and a baseline you defined, and that is an invoice nobody enjoys auditing.

IDEA 03

The bid machine.

Connect a company's past proposals, product docs, certifications, pricing, and past performance. Then monitor procurement portals continuously, read the 180-page solicitations, screen for eligibility, score fit, research the buying agency, build the compliance matrix, draft the response, assemble the attachments, check every requirement against every section, and stage the submission. A human reviews and presses submit.

It fits the filter because of an asymmetry. Reading and screening are enormous, mechanical work whose output can be verified automatically, since a compliance matrix is either right or wrong against the solicitation. Meanwhile the value of a single win runs to six or seven figures. Most companies bid on a small fraction of what they are eligible for, purely because qualifying costs days.

WHO BUYS

Mid-size government contractors, specialty subs, grant-dependent nonprofits

HOW IT PRICES

\$500–\$2,000/month, plus a success fee on wins

VERIFICATION COST

Split. Compliance checks cost nothing; persuasive quality costs a person.

THE RISK

This is the most crowded idea on the list. An established category of AI-first proposal tools already claims to cut response time by half to four-fifths, sitting alongside legacy content-library platforms that hold the incumbency. More fundamentally, the last 10% of a winning bid is genuine domain differentiation the model cannot invent, which leaves you selling assembly, and assembly commoditizes. Then there is the arms-race problem: if this works broadly it works for everyone, submission volume rises, and win rates settle back near where they started.

Three aimed at the interfaces that will never have an API.

This is where Astra’s actual improvement lands, and the improvement is patience rather than reasoning: the ability to sit in front of a 1998-vintage web form for forty minutes without losing the thread.

An enormous amount of white-collar labor is human beings acting as adapters between systems that refuse to talk to each other. That work survived every previous automation wave for one reason: integration required either an API that didn’t exist or brittle scripting that broke on the next UI release. A model that can look at a screen and adapt changes the failure mode from “breaks” to “costs a retry.”

IDEA 04

Filings as a product.

Contractor license renewals. State business registrations. Certificates of insurance. Residential and light commercial permits. Every jurisdiction has a different bad website, a different current PDF, a different submission channel, and a different reviewer with a different pet objection, and no one will ever build an API for any of it. Legal interpretation was never the hard part. The maze is the whole job: which municipality has jurisdiction, which form is current, what drawings are required, who gets emailed, why it was rejected, what expires next year.

Price per filing and start absurdly narrow, at something like *every residential EV-charger permit in New York*, then widen only after the jurisdiction knowledge compounds. Verification is unusually clean here: a submission either produces a confirmation number or it does not, and every rejection notice is labeled training data arriving free in your inbox.

WHO BUYS	HOW IT PRICES	VERIFICATION COST
Trades, small commercial builders, solar and EV installers, multi-state operators	\$199–\$999 per filing, which is what the manual version already costs	Near zero on submission. Higher on correctness, which arrives later.

THE RISK

The moat here is the jurisdiction dataset rather than the agent, and someone else has been accumulating one for five years. The best-funded player in construction permitting raised \$54M last December at roughly a half-billion valuation on the back of twelve million municipal data points and enterprise builders as customers. You would be starting that flywheel from zero. Two more specific hazards: many government portals explicitly prohibit automated submission and defend with CAPTCHAs, and a wrong filing carries real liability that lands on you rather than on the model vendor. At \$299 a filing, one bad one wipes out a year of the account.

IDEA 05

The integration shop, for everything too weird for Zapier.

A customer describes a workflow in one paragraph: *when the supplier portal gets an order, download the PDF, key these five fields into the ERP, upload tracking, and email Susan unless the customer is in Canada.* Today that is a grim \$20,000 consulting engagement, and it stays undone in most companies because \$20,000 is more than the annoyance is worth.

The agentic version inspects both applications, reads whatever documentation exists, watches the existing manual workflow, writes the integration, falls back to browser automation where there is no API, generates its own test cases, deploys, monitors, and repairs. The adjacent and possibly better-shaped version of the same capability is fixed-fee data migration: moving a company off one system onto another is 90% grunt work, and verification is nearly free because you can diff record counts and field-level checksums against the source.

WHO BUYS

Ops leaders at companies with old software and no engineering team

HOW IT PRICES

Setup fee plus monthly per workflow; fixed-price for migrations

VERIFICATION COST

Free for migrations. Ongoing and unbounded for live integrations.

THE RISK

You are building on top of someone else's UI, on their release schedule, forever. Every integration you sell is a permanent maintenance obligation priced as if it were a product, and the margin dies in support long before it dies in inference costs. Worse, the failures are silent and consequential: an order that quietly fails to sync gets discovered downstream by the customer, weeks later. The migration variant is the healthier business precisely because it ends. You deliver, verify against checksums, hand over, and leave. Anyone drawn to this idea by the recurring revenue has the economics backwards.

The synthetic user.

Instead of writing test cases, release the software to a thousand slightly deranged simulated customers and let them use it. Each one gets a persona and a set of bad habits. I don't understand accounting. I double-click everything. I hit Back constantly. I abandon checkout halfway. I upload a 300MB file. I have forty-seven thousand records. I've been a customer for seven years and my data is a mess.

They operate the actual application, wander into workflows nobody designed, break things, and return reproducible reports with steps, screenshots, video, and network logs. The product being sold is **millions of simulated minutes of software usage**, which is a different thing from automated testing. It is also the purest "throw more tokens at it" business in the document, because the output is objectively checkable by construction: a bug either reproduces or it does not.

WHO BUYS	HOW IT PRICES	VERIFICATION COST
Product and engineering teams shipping fast without QA headcount	Usage-based, per thousand testing hours, which aligns cost to compute	Zero. Replay the steps; it reproduces or it's noise.
THE RISK Roughly \$1.5B of venture money has already gone into AI testing, across forty-plus startups, several with real logos. You would be late. The economics are subtly hostile too: developers are cheap buyers who will try to build this with the model subscription they already have, and the marginal value of bug number 400 is close to zero, so teams drown in findings and start ignoring the feed, which looks exactly like churn from the outside. Survival in this category comes down to triage and signal quality, which is unfortunate given that raw volume is the thing that just got cheap.		

A service company with no employees and no dashboard.

Every other idea here is software sold to someone who then has to operate it. This one refuses to ship software at all, which makes it either the most durable item on the list or the least venture-shaped.

Pick one awful administrative workflow and sell the finished work. *We handle everything after a home inspector finishes an inspection. We run the back office for independent adjusters. We process supplier invoices for electrical contractors. We do the post-close file for small title agencies.* The customer never sees a product and never learns a prompt. They email `operations@yourcompany.com` and completed work comes back.

You are selling exactly what an outsourcing firm sells, which is delivered outcomes, except that the labor pool is tokens. That arbitrage is measurable rather than theoretical: traditional outsourcers run at roughly 25–30% gross margins, and AI-native operators claim blended margins around 64% at a 70/30 split of machine to human work. Treat that specific figure with suspicion, since it comes from people selling the model, but the direction gets corroborated by the incumbents' own behavior. One of the large consultancies paid \$3.3B for an outsourcing firm to buy its way in, and early adopters report operating-margin expansion of 750 to 1,000 basis points.

**The customer emails an inbox and gets completed work back.
Behind the inbox are thirty agents and one person who fixes what they break.**

The structural advantage is that a services wrapper absorbs the 41%. When the agent fails, a human on your side fixes it and the customer never knows, so the failure shows up as an internal cost line rather than a broken product experience. No other shape on this list has that property, which is why this is the one idea here that does not need the next model to be better.

THE RISK, AND IT IS THE ORDINARY KIND

It is a services business and it has services-business problems: linear scaling until you have built enough shared tooling, customer concentration, key-person dependency on whoever actually understands the workflow, and the constant temptation to accept adjacent work that breaks your automation. It gets valued as a services business too, at a multiple that is a small integer, with a strategic buyer or a dividend at the end rather than a Series C. The likeliest bad outcome here is a company that works fine and never gets interesting, which is both more probable and less discussed than anything else in this document.

The honest counterweight: it might be the best risk-adjusted item here. A boring \$10M service company with 60% gross margins is a genuinely good life, and it does not require anyone's AGI thesis to be correct.

PART SIX · WHAT HAS TO BE TRUE

Five conditions, and the objections without good answers.

Skipping this section is how a market map becomes a pitch deck. These are load-bearing: fail one and most of the list fails with it.

One. Verification has to be cheaper than production. This is the master condition and everything else is downstream of it. Where confirming the work requires the same expert whose time you were trying to save, the cost has moved rather than gone. Anyone who cannot state what a single check costs, in dollars and in seconds, has not yet found the business.

Two. Failure has to be cheap. A wrong test result costs a retry. A wrong permit filing costs a rejection, a delay, and possibly a liability claim. A wrong invoice dispute costs a supplier relationship. A wrong tax position costs a client. The same 41% success rate is a scheduling detail in one column and an uninsurable exposure in the next, and the ideas in this document are not evenly distributed across those columns.

Three. The work has to decompose into units inside the reliability horizon. Frontier agents are near-perfect on sub-four-minute tasks and under 10% on tasks that would take a human more than four hours, with errors correlating across steps rather than accumulating independently. The projection is that the 50% reliability horizon reaches roughly four hours around 2027. Building today for the horizon you expect in eighteen months is the most common and most expensive error in this space.

Four. Someone has to already be paying for the bad version. Every idea in this document that I'd actually fund replaces an existing invoice: a recovery audit firm, a permit expeditor, a proposal consultant, an offshore back office. Net-new demand for work nobody currently buys is a much longer road, and "we made this so cheap that people who never wanted it will now buy it" is a thesis with a poor historical record.

Five. The moat has to be somewhere other than the model. Everyone gets the same Astra on the same Tuesday. Defensibility comes from proprietary data, workflow-specific evaluation harnesses, integration surface, regulatory standing, or distribution, and given the 37-point harness gap the scaffolding is a more real asset than it sounds. Anything a competitor can reproduce by writing a better prompt will be reproduced inside a quarter.

Five objections come up whenever this is presented, with the most honest response I have to each. One of them I have no answer to at all.

THE OBJECTION	THE HONEST RESPONSE
41% success is a toy. You cannot run a business on a worker that fails six times out of ten.	Good answer available. You never ship raw output. Attempt, check, retry, escalate; the halved hallucination rate is what makes the check cheap. The metric that matters is cost per <i>accepted</i> unit, and nobody quoting 41% has computed it.
Tokens went the wrong way. Astra is 2.5× the price and measured cost per task went up.	Correct. The standard rebuttal is that the price of a fixed capability has fallen on the order of 10× a year, which is a trend line rather than a budget. Expect 3–5× annual declines near-term, then a taper. Contingency and outcome pricing hedge the exposure; fixed-price seat SaaS leaves you holding it.

THE OBJECTION

THE HONEST RESPONSE

Every one of these already has a funded competitor. Permitting, RFPs, QA, revenue recovery, all of them.

No good answer, and this is the strongest objection in the document. The idea was never scarce; the wedge is a specific vertical, a specific dataset, a specific regulatory permission, or a specific buyer relationship. If your only edge is "we use the new model," you have a six-week lead on people with \$54M.

OpenAI just ships this. The model vendor absorbs every thin wrapper eventually.

Partly answerable. Vendors ship horizontal capability and stop short of vertical accountability; they will not assume liability for your permit filing or chase your supplier for a refund. What gets absorbed is whatever a customer could have prompted for themselves.

The headline capability isn't real. The 99.9% was a harness, general intelligence is flat, and GDPval went backwards.

All true, and it barely matters for these seven. None of them need general intelligence. What they need is long-horizon computer use, which is the one thing that unambiguously improved.

Scored against all of that, the list separates cleanly. The middle column is the one that decides whether you have a business; the right column is what you're betting on.

IDEA

VERIFICATION COST

THE THING THAT HAS TO BREAK YOUR WAY

01 · Recovery agent

Near zero

You can collect as well as find, and the small accounts add up

02 · Exhaustive sourcing

Low

Suppliers don't gate you out once it works

03 · Bid machine

Mixed

You find a vertical the incumbents haven't taken

04 · Filings as a product

Low, delayed

You build jurisdiction data faster than the funded incumbent, and survive the liability

05 · Integration shop

Free for migrations, unbounded for live

Maintenance doesn't eat the margin. Migrations dodge this entirely.

06 · Synthetic user

Zero

Triage quality beats volume, in a category that's already crowded and funded

07 · Zero-employee services

Internalized

Nothing, really. This works at today's capability and today's prices.

Idea 07 is the outlier and it's worth saying plainly. It needs no capability improvement, no price decline, no access tier you can't get, and no thesis about AGI, only a boring workflow and a willingness to run an unglamorous company. It is the one item here that survives if the argument in Part two turns out to be wrong. Idea 01 has the largest ceiling, available only to whoever builds the collections half alongside the search half.

THE PART THAT ISN'T SPECULATIVE

Four things are true regardless of where you land on the AGI question. Long-horizon computer use measurably improved and the improvement is large. Hallucination rates roughly halved, which changes verification economics more than any capability gain. Enormous volumes of white-collar work are humans acting as adapters between systems that won't talk. And buyers are already spending real money on the manual versions of all seven of these. A business that pencils under those four facts alone, with further model progress as upside rather than as the plan, is a much better-shaped bet than one that needs next year's model to arrive on schedule.

PART SEVEN · SIGNPOSTS AND A CHEAP TEST

You can find out whether yours works in about thirty days.

For considerably less than the cost of being wrong about it for a year. The point is to replace the market map with evidence you gathered yourself.

The indicators first. These would move this from an interesting capability to a settled market, roughly in the order they would have to happen.

IF YOU SEE THIS	IT MEANS
Independent harnesses close the gap on the headline benchmarks	The capability is in the weights and portable. Right now a lot of it is in scaffolding, which is both an opportunity and a warning.
Token prices resume falling at the frontier tier	The volume-heavy ideas, 02 and 06 especially, get their margins back. Until then, price on outcomes.
A published reliability horizon past two hours on real work	Decomposition stops being the whole architecture, and much larger workflows become sellable as single units.

IF YOU SEE THIS

IT MEANS

Someone posts an audited cost-per-accepted-unit number

The discourse gets a floor and you can do arithmetic instead of estimates. Watch for an audited figure rather than a case study.

Government portals start systematically blocking agents

Idea 04 gets much harder and much more valuable at the same time, because the barrier is now a moat.

An insurer writes a policy covering agent-caused errors

Underwriters have priced the failure rate, which is the precondition for any of this touching regulated work.

A traditional BPO discloses its automation ratio

The margin claims in Part five become checkable. Right now they're vendor marketing.

The test second. For anyone seriously considering one of these, this sequence produces better information than any amount of market sizing, and it fits in a month of evenings.

WEEK ONE

Time the human doing it, before you automate anything.

Find one person doing the actual workflow and measure it. Minutes per unit, error rate, what they look up, where they stall, what they escalate, and what it costs when they get it wrong. You are establishing the denominator. Nearly everyone in this space skips straight to prototyping and consequently has no idea whether their agent is cheaper than the thing it replaces.

WEEK TWO

Build the checker first.

Before building anything that does the work, build the thing that decides whether the work is correct, and build it without the model where you can, because a checker sharing the doer's blind spots will pass everything. If no cheap mechanical acceptance test exists for a unit of output, you have your answer a week in rather than two quarters in.

BE HONEST ABOUT WHAT THIS PROVES

Twenty hand-picked examples passing does not make a checker. Salt the set with known-bad units and confirm it catches them. Most verification layers in this category are measuring agreement rather than correctness, and the difference does not show up until a customer finds it.

WEEK THREE

Run a hundred real units and compute one number.

Not synthetic ones. A hundred real units from a real customer, end to end, at your real retry policy. Then compute cost per *accepted* unit, with everything in it: retries, the failed attempts you threw away, human review minutes at a loaded rate, and rework. Compare it to week one's number. If it isn't at least 5× better, the margin will not survive first contact with support costs and edge cases.

WEEK FOUR

Write down what would make you stop.

Name the specific result that kills it before you're emotionally invested: cost per accepted unit above a threshold you set in advance, a verification step that turns out to require an expert, a liability exposure you can't insure, or a funded incumbent shipping your wedge. Write the numbers down. People rarely lose three years in an emerging market by picking the wrong idea. They lose it by never defining disconfirmation and then reading every ambiguous result favorably.

IF THE THIRTY DAYS COME BACK NEGATIVE

Being early only pays if the capability arrives on your schedule. Most of the money in previous platform shifts went to people who were early by about eighteen months rather than five years. Where the reliability isn't there yet, the usual move is the services version, idea 07, which absorbs the failure rate internally and pays you while you wait for the number to move.

A real capability shipped, in a narrower band than the coverage suggests. Long-horizon computer use got substantially better and substantially faster, and hallucination roughly halved. General reasoning did not move, one economically-relevant benchmark went backwards, the flagship number depended on a harness, and tokens got more expensive.

The businesses that work in this window are the unglamorous ones: somebody already pays for the manual version, the output can be checked by a machine for pennies, the work decomposes into short units, and being wrong is survivable. Most of the seven above fail at least one of those, and which ones fail depends on details of your own workflow that no market map contains. The thirty days in Part seven cost less than one bad quarter.

ONE MORE THING

If you have a workflow in mind and want a second read on the arithmetic.

Which is, admittedly, weeks one through three of the test performed by somebody who isn't you.

The most interesting thing about this whole shift is that almost nobody arguing about it has done the division. Cost per accepted unit against the loaded cost of the human currently doing it: that single number settles more of this than a week of thinkpieces, and it comes out different for every workflow.

I'm doing a small number of informal reads. Send me the workflow you're thinking about, including what it is, who does it now, how long it takes them, and how you'd check the output, and I'll tell you which of the seven shapes above it fits and whether the verification step is cheap enough for it to be a business. No pressure; everything here is yours to use either way.

GET IN TOUCH

Email hi@davecto.com with the subject line **"Agent Economics"** and a couple of sentences about the workflow.

More guides like this one, for people trying to use AI without embarrassing themselves.

Weekly, plain-language breakdowns on Instagram.

[@davecto](https://www.instagram.com/davecto)

A note on sourcing: benchmark figures come from OpenAI's launch materials and from independent evaluators (Artificial Analysis, ARC Prize, and the Agents' Last Exam leaderboard), and where those disagree I've given both numbers rather than the flattering one. The reliability findings on long-horizon agents come from recent academic work on error correlation and task-length scaling. Market and funding figures come from press coverage and, in the case of the outsourcing margin numbers in Part five, from vendors with an obvious interest in them; treat those as directional. Benchmark scores in a week-old model release move fast, and access tiers were still shifting while I wrote this, so verify anything you plan to build

on. The seven ideas, the scoring, and the thirty-day test are argument rather than research. I have no inside information about any company named here, no revenue figures from anyone in a position to know them, and no way to check the Part five margin claims against an audited filing, because none has been published.