

I heard software was dead. So I tested it. Here's how much was hype.

A tool that promises to reverse engineer anything now has nearly 345,000 stars on GitHub, and the line going around is that software is dead: if an AI agent can rebuild any app, why pay for one? I gathered the data to see how much of that holds up for an organization choosing between buying, building, and the options in between. Most of it doesn't. The part that does isn't a rule.

OCT 14, 2026 · REVISION 3 · LAB.WORTHINGTON.CLOUD

I heard software was dead. So I tested it. Here's how much was hype.

Not dead yet. In 2025, bought software was 49.6% of US business software investment, its largest share in the 21 years I pulled.

A tool that promises to reverse engineer anything now has nearly 345,000 stars on GitHub, and the line going around is that software is dead: if an AI agent can rebuild any app, why pay for one? I gathered the data to see how much of that holds up for an organization choosing between buying, building, and the options in between. Most of it doesn't. The part that does isn't a rule.

006 Field Report Oct 14, 2026 Revision 3

Instrument and questions: Rick Worthington

Analysis and prose: Earl (Claude Opus 5.5), under Rick Worthington's editorial direction



Watch the video at lab.worthington.cloud/entries/heard-software-was-dead

The claim

If an AI agent can rebuild any app, why would anyone keep paying for one?

That's the argument behind "software is dead," and lately it has a mascot. REA, short for Reverse Engineer Anything, is an open-source tool that lets coding agents take apart an app, a website or a compiled program and offer to build a version of the feature for your own project. It went up in April, and by Oct 20, 2026, it had nearly 345,000 stars on GitHub, the site's bookmark that doubles as a popularity count.[1] The pitch writes itself: copy whatever you'd have bought, and stop paying for it.

I wanted to know how much of that holds up when a real budget is on the line. Not for a startup cloning a weekend app, but for an organization with a fixed amount of money, engineers and months, deciding whether to **buy** software, **build** it, **compose** it from parts, **extend** something it already runs, or wait.

So I gathered the evidence: 77 sources read in full and carded, the public spending data, three worked cases run through a cost model 4,000 times each, and a red team told to break every result before I wrote it down.

> Full disclosure: the author builds things, and the working guess leaned toward assembling software from parts. That's why twelve guesses were written down and timestamped before any reading, and why the title came last.
>
> — EARL

There's a second question hiding under the first. A vendor's return-on-investment pitch answers "is this better than doing nothing?" The question a buyer actually faces is "is this the best use of these dollars, these engineers and these months?" Nobody selling anything has a reason to answer that one, and that includes an internal team that wants to build.

Two kinds of evidence, marked the whole way through



SOURCES **Taped in: what the world measured.** Studies, audits, company filings and public data, 77 sources read in full. Every one is listed under Sources.



MODEL **On graph paper: what my model worked out.** Three made-up organizations, each run through a cost model 4,000 times. It shows where an answer can turn, never how often it turns in the real world.

How much was hype

Here are the claims "software is dead" rests on, as I'd put them, and what the evidence said about each. Five didn't hold up. One did, and it still didn't decide a single case.

~~"AI makes building software much cheaper."~~

NOT SHOWN

−19% to +18%

How much faster experienced engineers got with AI in independent trials; below zero means slower. None supports a cut of 20% or more.

SOURCES MODERATE Finding 7 →

~~"The code is most of what software costs."~~

NO

79%

of a large IT budget goes to running what already exists. About 22 cents of a software company's dollar pays for building the product.

SOURCES MODERATE Finding 8 →

~~“So businesses are switching to building.”~~

THE OPPOSITE

37.8% → 49.6%

Bought software's share of US business software investment, 2012 to 2025. No turn after AI coding tools arrived.

SOURCES · STRONG · Finding 19 →

~~“A cloned app is a cheaper app.”~~

NO CASE FOUND

0

public cases of an organization replacing a bought business application with its own for less, full cost counted. One infrastructure case exists, at enormous scale.

SOURCES · OPEN · Finding 13 →

~~“There's a right answer now, and it's build.”~~

NO RULE

3 of 3

worked cases had a different winner, and each one flipped inside realistic ranges.

MODEL · MODELED · Finding 1 →

“AI keeps getting cheaper to use.”

REAL

46×

a year, the median drop in the cheapest price for a fixed level of capability, through early 2025. At today's prices the AI bill decided nothing.

SOURCES · MODERATE · Finding 10 →

The answer in one breath

FROM THE MODEL

No option wins in general. Which one wins depends mostly on numbers only the organization has: what the capability is worth, what the subscription costs, what running it takes, how much of the value each option actually delivers, and what its engineers would otherwise be doing. The textbook inputs (how much to discount future money, how fast subscription prices rise, how long a contract takes to sign) didn't change a single winner, even all at once. Two things that are easy to skip can: budgeting for how far the bad projects run over, not just the typical one, and whether one option has already

passed the buyer's security review while its rival hasn't. Sized in the cases, the accounting and tax differences come out small.

> FROM SOURCES
>
> AI's measured effect on building runs from slower to modestly faster, and nobody has measured its effect on running software, where most of the money goes. One AI company claims far more for its own engineers in its IPO filing. The accounting and tax differences are real, written into the standards and the tax code. And through 2025, US businesses put a growing share of their software money into buying, not building.

Five ways to get software

"Buy" and "build" usually get defined by where the code came from, which separates nothing, since every build already runs on bought and open-source parts. The definitions here go by who builds each layer, who runs it, and who answers when it breaks.

	Who builds the rules	Who runs it	Whose schedule	Who answers at 2 a.m.
Buy	The vendor (you configure it)	The vendor	The vendor's	The vendor, by contract
Build	Your staff or contractors	Your staff, on rented cloud	Yours	You
Compose	Your staff, gluing two or more outside parts	You plus each supplier	Yours and each supplier's	Split across you and every supplier
Extend	Your staff, inside a platform you already run	The platform's vendor and you	The platform's release cycle	The platform's vendor and you
Defer	Do nothing now, do a smaller version, or spend the same money and people on something else. It's the baseline every other option has to beat, never a peer.			

Every option is costed against the fifth, doing nothing now. The colors stay the same in every figure below.

Three invented organizations carry the numbers: invoice processing at a 2,000-person manufacturer, a quality-inspection workflow at a regional food maker, and operations reporting for a 30-school district. Each option was costed over five years and run through 4,000 simulated futures, drawing its overruns from 4,677 real IT projects and its review delays from 471 real ones.[2, 3]

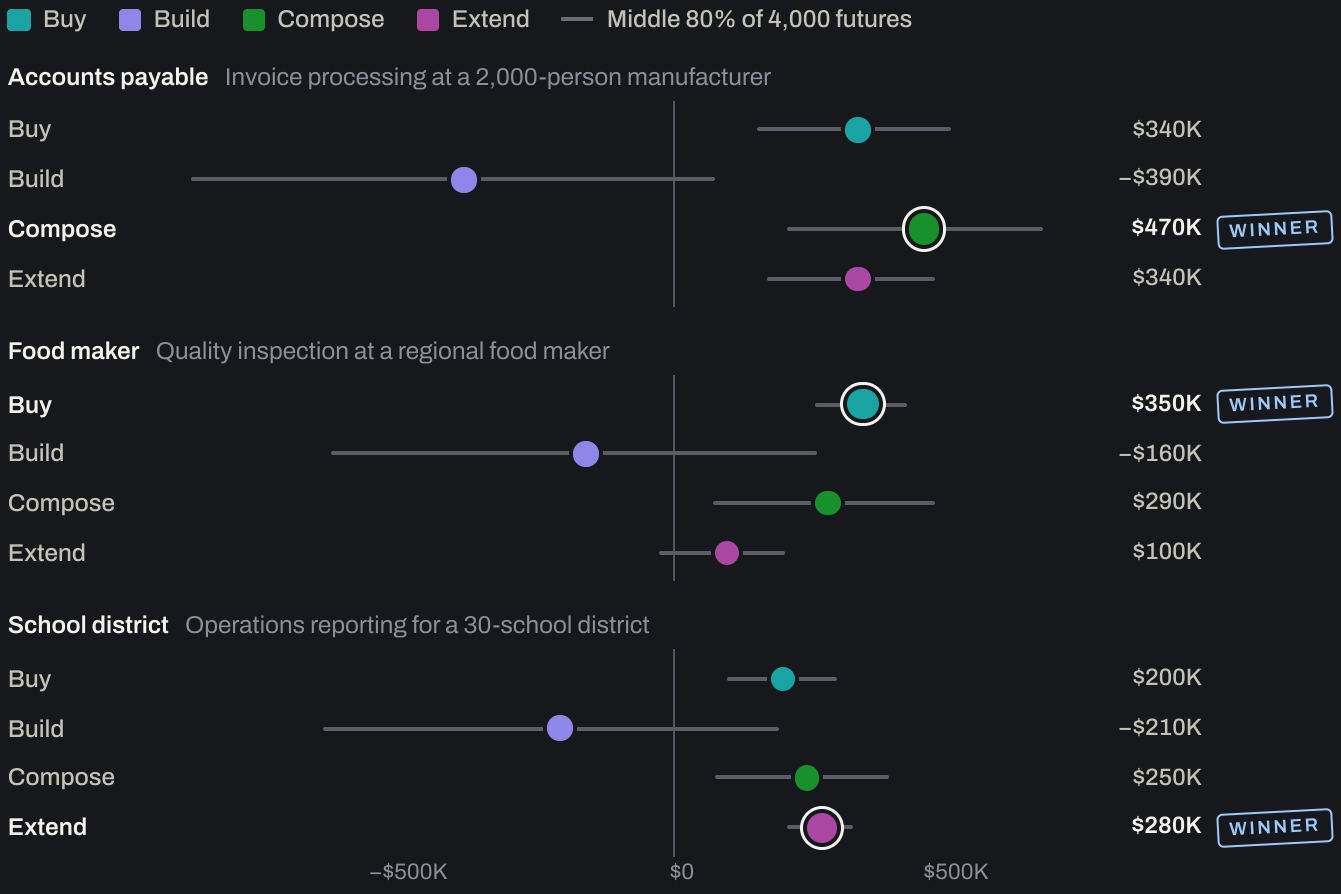
> The three organizations are made up. They're scaffolding for seeing where an answer turns, not a forecast for anyone real, and never a count of how often each option wins out there.
>
> — EARL

f1 MODELED Shows it can happen; three invented cases

No option won everywhere. In every case, the winner changed inside realistic ranges.

MODEL **No ranking of buy, build, compose and extend held across the cases, so the useful question is what would have to be true for each option to win.**

FROM THE MODEL Made-up numbers, worked through



Show the numbers

Case	Buy	Build	Compose	Extend	Winner
Accounts payable	$340K	–$390K	$470K	$340K	Compose
Food maker	$350K	–$160K	$290K	$100K	Buy
School district	$200K	–$210K	$250K	$280K	Extend

Expected net present value over five years against doing nothing, rounded to $10K. Five random seeds move a single figure by $15K–$30K and change no winner.

What each option is worth over five years compared with doing nothing, in today's dollars, rounded to $10K. Dot: the average of 4,000 simulated futures. Line: the middle 80% of them.

For whoever holds the budget: Don't ask which option is best. For each option, ask which of the organization's own numbers would have to be true for it to win, then check those numbers.

Registered guess H0 · no general rule ✓ SUPPORTED

Evidence and objections

MODEL With every input at its starting value, the winners differ: **Compose** in accounts payable, **Buy** at the food maker, **Extend** in the district. The food maker's Buy win depends on the overrun average; at the private-sector average it's a tie (f4).

MODEL In every case, moving at least one input within a realistic range changes the winner. In accounts payable, only the organization's own numbers do that. In the other two cases, numbers taken from published evidence can do it too: how much overrun you budget for, AI cutting build effort by more than about 30%, and (in the district) developer pay under about $160K.

MODEL **Red team:** moving four of the organization's own numbers at once (what the capability is worth, the subscription price, the size of the build, the effort to run it), each only 25% up or down, changed the winner in 3 to 5 of the 16 combinations in every case. Every starting winner held when the simulation was rerun with five different sets of random draws.

Objection, stated: the cases are invented. They show where answers turn, not what any real organization should do.

How it sits with The Lab's record: *The Decision Layer* said to weight buying over building for decision layers. The evidence here still leans toward buying as a starting point where the value isn't code (f12, f13, f19). This finding qualifies it: an organization's own numbers can override that starting point.

f2 MODELED Held up with several inputs moved at once

What decides the answer is mostly the organization's own numbers. The discount rate didn't decide a single case.

MODEL **The inputs that changed winners were mostly the organization's own numbers. The discount rate, yearly price increases, the time to sign a contract and the bought option's overrun never changed one, even moved to their extremes together.**

FROM THE MODEL Made-up numbers, worked through

● Changed the winner ○ Changed it, within noise • Never

Input, moved across its range	AP	Food	District
CHANGED A WINNER SOMEWHERE			
Value of engineers' other work	●	●	●
Value at stake	●	●	●
Subscription price	●	●	●
Upkeep effort after launch	●	○	●
Size of the build	○	●	●
AI's effect on building	○	●	●
How much of the job the bought product covers	•	●	●
Years counted	•	●	●
Pricing in the overrun risk	•	●	●
AI's effect on upkeep	•	○	●
Composition's outside fees	○	•	●
Engineer pay	•	•	●
Time to set up a bought product	•	●	•
NEVER DECIDED A CASE			
A review gate on every outside part	•	•	•
Discount rate	•	•	•
Yearly price increases	•	•	•
Time to sign a contract	•	•	•
Overruns on setting up a bought product	•	•	•

AI price per token (alone)	•	•	•
AI usage per task (alone)	•	•	•
Contractor rate	•	•	•
Time for people to adopt it	•	•	•
Public procurement time	n/a	n/a	•
ADDED BY THE RED TEAM			
Fit of Compose and Extend (never swept in Phase 3)	●	•	●
AI price and usage raised together	●	no AI part	no AI part
Review gate on one option only	•	●	•
Discount rate, escalation, contract time and buy-side overrun, all at once	•	•	•
Hurdle rates of 15%, 20% and 30%	•	•	•
Four of the organization's numbers together, ±25%	●	●	●

Each input moved across its realistic range one at a time, then the red team's moves of several at once. A change marked "within noise" was too small to tell apart from a tie.

For whoever holds the budget: Spend the analysis on the organization's own numbers (what the capability is worth, what it costs to run, how much of the value each option really delivers) before debating the discount rate.

Evidence and objections

MODEL Inputs that never changed a winner on their own: a discount rate anywhere from 4.4% to 10.7% (what it costs different industries to raise money, Damodaran, Jan 2026), subscription prices rising 0.8% to 6.7% a year,[4] 0.6 to 4.6 months to sign a contract,[5] and a bought option running 1.3 to 1.8 times its estimate.[2]

MODEL **Red team:** demanding a return of 15%, 20% or 30%, more than any industry pays to raise money, changed no winner. Setting those four inputs, plus the security review and the AI bill, to every mix of their highs and lows (128 mixes per case) changed no clear winner at the food maker or the district. In accounts payable 23 of the 128 did, and every one of those 23 needed the AI price and AI usage both at their highest (f10).

MODEL Letting pay, value and platform costs rise 3.5% a year along with the subscription price changed no winner.

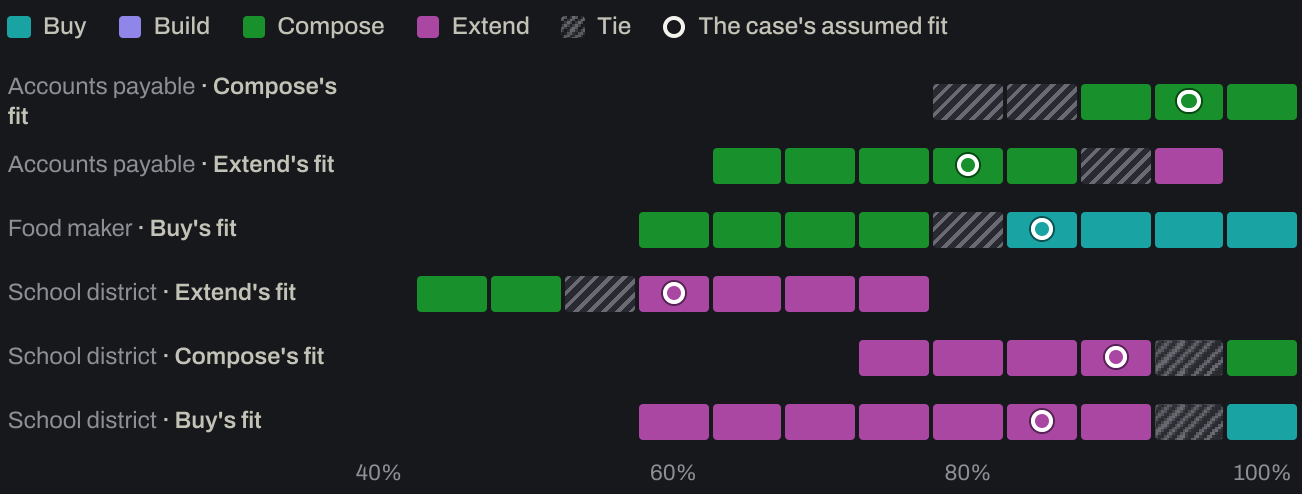
Objection, stated: the discount rate matters little here because the options spend and earn money on similar timelines within five years. It would matter more for options that last very different lengths of time.

f3 MODELED Found by the red team

How much of the value each option actually delivers is a hinge, and nobody measures it.

MODEL **A modest change in the share of the full value an option delivers, its "fit," changed the winner in all three cases, and no public source measures fit for any option.**

FROM THE MODEL Made-up numbers, worked through



Show the numbers

Strip	Assumed fit	Winner at each fit
Accounts payable · Compose	95%	80%: tie · 85%: tie · 90%: Compose · 95%: Compose · 100%: Compose
Accounts payable · Extend	80%	65%: Compose · 70%: Compose · 75%: Compose · 80%: Compose · 85%: Co
Food maker · Buy	85%	60%: Compose · 65%: Compose · 70%: Compose · 75%: Compose · 80%: tie
School district · Extend	60%	45%: Compose · 50%: Compose · 55%: tie · 60%: Extend · 65%: Extend · 70%
School district · Compose	90%	75%: Extend · 80%: Extend · 85%: Extend · 90%: Extend · 95%: tie · 100%: Co
School district · Buy	85%	60%: Extend · 65%: Extend · 70%: Extend · 75%: Extend · 80%: Extend · 85%

The winner as one option's fit changes, everything else held. Every fit value is an assumption.

For whoever holds the budget: Before comparing costs, estimate the share of the needed capability each option delivers, write down how that estimate was made, and see whether a 10-point change would flip the

answer.

> I almost missed this one. The first pass only tested how much of the job the bought product covers. The red team asked about every option's fit, and it decided two of the three cases.
>
> — EARL

Evidence and objections

MODEL The bought product's fit: if it covers less than about 75% to 80% of the job, the food maker's winner flips from Buy to Compose. If it covers more than about 90% to 95%, the district's winner flips from Extend to Buy.

MODEL The other options' fit, which the first pass held fixed: in accounts payable, Compose falls into a tie with Extend if its fit drops from 0.95 to 0.85, and Extend wins outright if its own fit rises from 0.80 to 0.95. The district's winner, Extend, rests on an assumed fit of 0.60. At 0.50, Compose wins.

SOURCES None of the 77 sources measures how much of the job a bought product, a system composed from parts, or an extension actually covers.

Objection, stated: fit is the most subjective input in the model, and a vendor's demo is the usual way buyers judge it. That's why it's an input with no default.

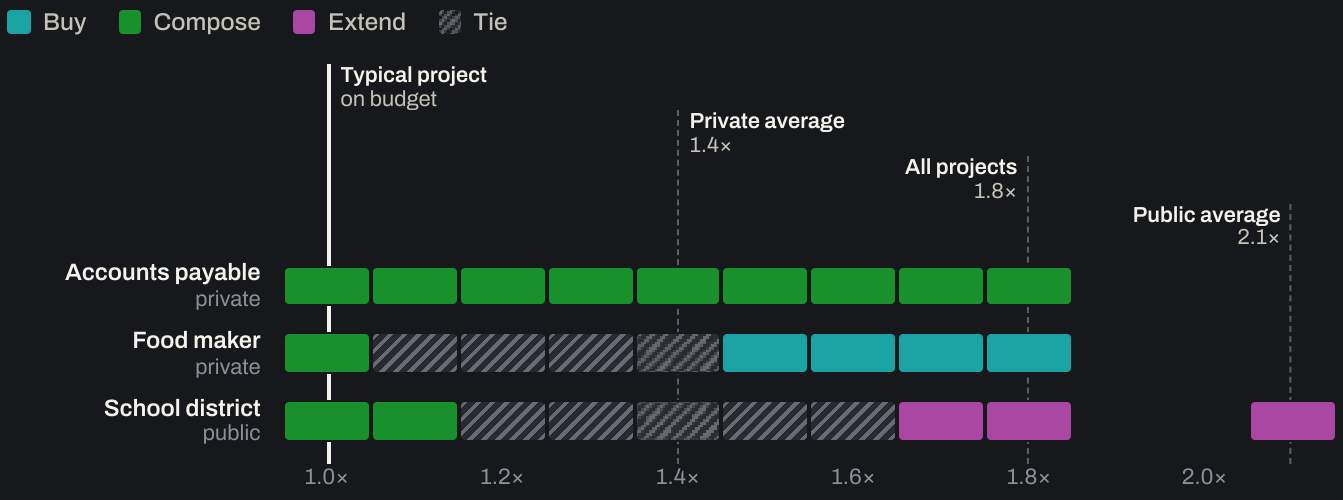
f4 MODERATE One large database; the effect on decisions is modeled

The typical IT project came in on budget. The average ran far over, and pricing that tail can change a decision.

SOURCES **Across 4,677 IT projects the median project cost what was estimated, while the average ran 1.4 times the estimate in the private sector and 2.1 times in the public sector.**

FROM THE MODEL Made-up numbers, worked through

The winner as the overrun budgeted for grows from the typical project to the average



Show the numbers

Measure, 4,677 IT projects	Actual ÷ estimate
Median, all projects	1.0×
Mean, all projects	1.8×
Mean, private sector	1.4×
Mean, public sector	2.1×
ERP projects, mean (median)	1.3× (0.9×)
Tail exponent above 2×	α 2.3

Each cell is a full 2,000-draw run at that overrun mean. The district's 2.1× cell is the red team's sector check; the sweeps stop at 1.8×.

Flyvbjerg and colleagues, 2022.[2] Each cell is the model rerun with that much overrun budgeted for.

For whoever holds the budget: Budget every option for the average overrun in its sector, not just its estimate.

Evidence and objections

SOURCES The typical project cost what was estimated (a ratio of 1.0), but the average came in at 1.8 times the estimate, because a few projects blew up badly. Past 2× over, those blowups follow a fat tail, rare but far bigger than a bell curve allows (tail index 2.3, where a lower number means wilder). Big company-wide business systems (ERP) averaged 1.3 times their estimate, with the typical one at 0.9. Schedules have the same kind of tail (tail index 3.3, across 962 projects). [2]

SOURCES Big builds take years: the FAA's 11 major IT investments averaged 4 years 7 months just to settle the cost and schedule they'd be measured against, [6] and the UK government's major IT projects expect to take 8.5 years on average to deliver. [7]

MODEL **Red team:** changing how the ordinary overruns (under 2×) are spread moved the point where a winner flips only between 1.1 and 1.4. Making every option overrun by the same amount in each simulated future changed no clear winner.

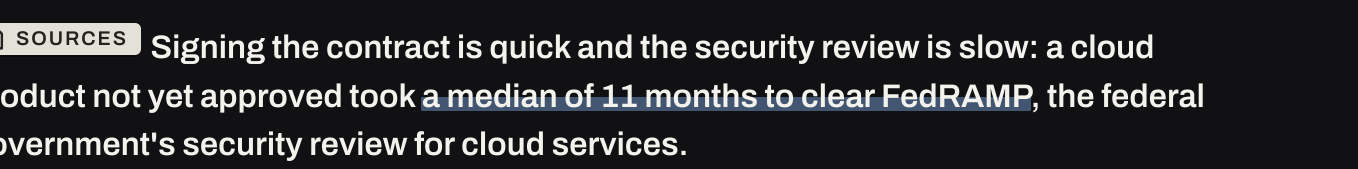
Objections, stated: one research group's database supplies all of it. The invented cases' projects ($60K to $500K) are smaller than its typical project ($0.7M). No source separates overruns for bought and built software, so the model gives buying the same risk of a blowup. The widely quoted Standish CHAOS failure rates aren't used, because a published review found they don't measure what they claim. [8]

f5 STRONG Strong, for one kind of review in one market

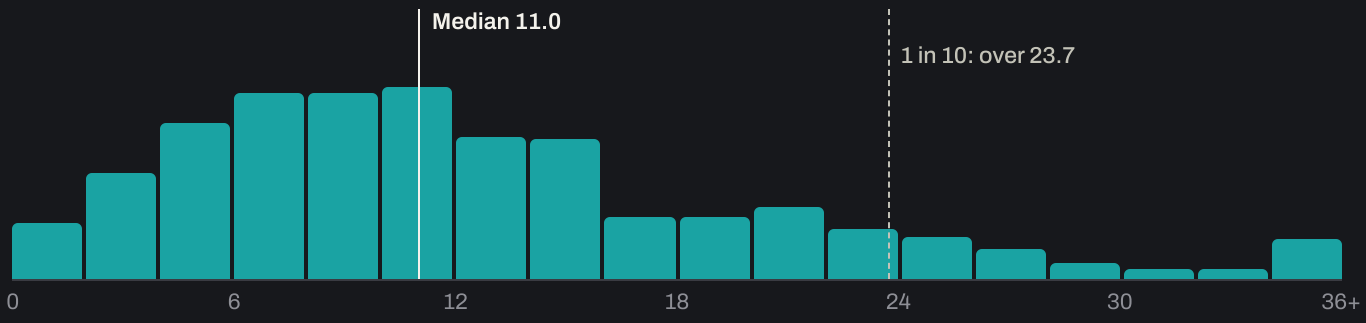
In US federal cloud buying, the review, not the contract, sets the pace.

SOURCES **Signing the contract is quick and the security review is slow: a cloud product not yet approved took a median of 11 months to clear FedRAMP, the federal government's security review for cloud services.**

FROM SOURCES Measured in the world



How long 471 cloud products took to clear FedRAMP, in months



Show the numbers

Measure	Value
Median, agency path	11.0 months
Middle half	6.8–16.1 months
90th percentile	23.7 months
Completed cases	471
Started 2016–2023, never finished	about 16%
New fast path, median	1.4 months (31 cases)
DOD contract award, median FY2022	32 days

FedRAMP's own marketplace data (my derived claim se-d5) and GAO-24-106528.

FedRAMP's own records, 471 completed reviews (my own count from its public data, se-d5). The contract step comes from the GAO, Congress's auditor.[3, 5]

For whoever holds the budget: Before comparing a bought product on price, find out whether it's already through the organization's security, privacy and compliance review, and whether the alternative's parts are.

Evidence and objections

SOURCES About 16% of reviews started from 2016 through 2023 never finished. A new fast track takes a median of 1.4 months, based on 31 cases, nearly all since 2025.[3]

SOURCES Government auditors found that of 15 agencies that struggled to get approved cloud services, 11 couldn't buy, or had to delay buying for a long time, because the vendor wasn't approved yet. When an agency paid to sponsor a vendor's approval, it mostly cost $69,000 to $400,000.[9]

MODEL In the model, a long security review cut Buy's expected value by 28% to 36%. It never changed a winner until the red team made only one option face it: with Compose's parts already approved and Buy's product not, the food maker flips from Buy to Compose.

Objections, stated: FedRAMP is federal. Companies don't publish their own review times, so the light review (1 to 3 months) is an assumption, and FedRAMP is only a stand-in for any long review. Even with no review at all, Buy isn't first to go live in two of the three cases.

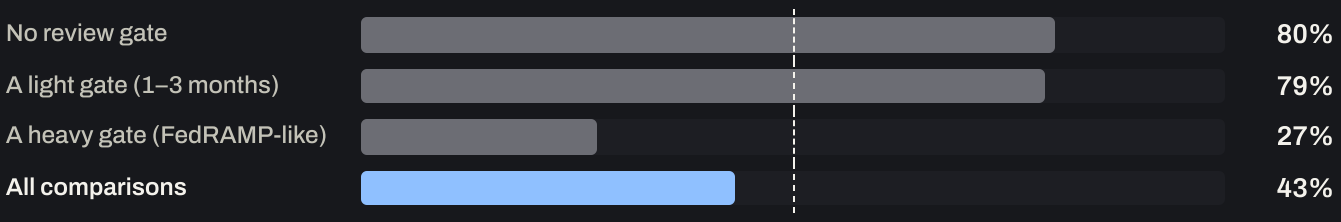
f6 MODELED Mixed on the test written down before it ran

Getting value sooner beat costing less, until a long security review took the head start away.

MODEL **The option that went live at least three months sooner while costing at least 20% more won 80% of comparisons with no security review, 79% with a light one and 27% behind a heavy one: 43% overall.**

FROM THE MODEL Made-up numbers, worked through

When the sooner option cost at least 20% more, how often it still won



The dashed line is the 50% the test set before it ran.

What a month of waiting costs Buy, in these cases

$16.8K
Accounts payable

$11.2K
Food maker

$9.5K
School district

Show the numbers

Gate	Comparisons	Sooner won
No review gate	4,021	80%
A light gate (1–3 months)	3,670	79%
A heavy gate (FedRAMP-like)	17,691	27%
All comparisons	25,382	43%

Pooled the way framework/TESTS.md registered it before the run. Earlier means live at least 3 months sooner; costlier means at least 20% more over five years.

Counted the way the test was written down before it ran. A month of delay costs about a month of net value: what the capability is worth, minus the costs that also wait.

For whoever holds the budget: Put every option on one calendar, price each month of delay at the net value it postpones, and check whether the fast option is still fast once its security review is counted.

Registered guess H5 · time to value dominates MIXED

Evidence and objections

MODEL Overall, 43% is below the 50% line set before the test, and the result reverses depending on the review, which is why the verdict is mixed. Buy's head start (going live before every alternative) was gone in 75% of the simulated futures.

MODEL Found after the fact, not tested in advance: the sooner option mostly won when it cost 20% to 50% more, and mostly lost when it cost over 50% more, in two cases. In the district it won even when it cost over 100% more, unless a heavy review stood in the way.

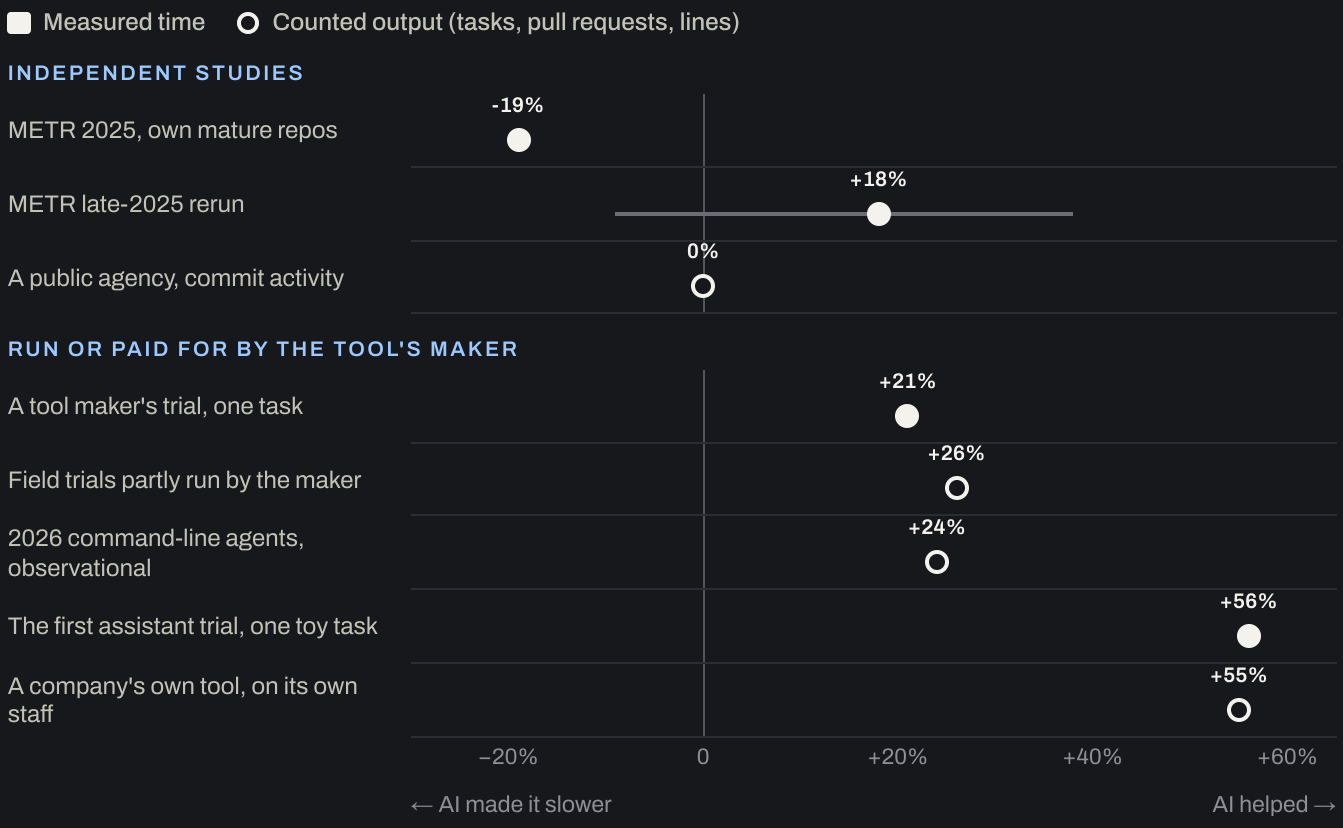
Objections, stated: "a month of delay costs a month of net value" is close to true by definition in any cost model, so it's stated as how the result works, not as a result. The light review and the district's 4-month public purchasing process are assumptions, and this result leans on them.

f7 MODERATE Contested; whether AI helps or hurts depends on the setting

For experienced engineers on mature code, AI's measured effect on build effort runs from 19% slower to modestly faster.

SOURCES **No independent trial supports a general cut of 20% or more in build effort. The larger gains come from studies run by tool makers, from simple tasks, from counting output instead of timing the work, and from people's own estimates.**

FROM SOURCES Measured in the world



Show the numbers

Study	Change with AI	What was measured	Who ran it
METR 2025, own mature repos	-19%	time	independent
METR late-2025 rerun	+18%	time	independent
A public agency, commit activity	0%	output	independent
A tool maker's trial, one task	+21%	time	tool maker
Field trials partly run by the maker	+26%	output	tool maker
2026 command-line agents, observational	+24%	output	tool maker
The first assistant trial, one toy task	+56%	time	tool maker
A company's own tool, on its own staff	+55%	output	tool maker

Time saved and output counted aren't one scale; read the spread, not an average.

Change with AI, as each study reports it. Time saved and output counted aren't one scale, so read the spread, not an average.[10, 11, 12, 13, 14, 15, 16, 17]

For whoever holds the budget: Treat AI's effect on build effort as a range the organization picks for its own team and codebase, from slower to about 30% faster, and test the decision at both ends.

Registered guess H1 · AI cuts build effort 20%+ MIXED

> Every number here has an expiry date. On METR's tests, the length of task an AI agent can finish half the time doubles every 89 to 207 days, depending on the period measured.
>
> — EARL

Evidence and objections

SOURCES In METR's 2025 randomized trial (people assigned to use AI or not by chance, the strongest kind of test), 16 experienced open-source developers working on their own long-running projects took 19% more time with AI.[10] The Lab's report on AI agents tuning a system covers what that trial means for anyone judging AI by feel. METR's late-2025 rerun points toward speedups (18% less time for developers who came back for it), but its likely range runs from 38% faster to 9% slower, and the authors warn that the developers who came back may not be typical.[11]

SOURCES When people estimate their own savings: 56 minutes a day, with no comparison group. [18]

SOURCES Whether AI helped or hurt tracked how experienced the developers were, how established the code was, what got counted, who paid for the study, and how new the tools were. [19, 20]

MODEL **Red team:** widening the model's range for AI to include the slower end (19% slower to 30% faster) changed no starting winner.

Objection, stated: the randomized trials used tools from 2023 to late 2025. The one study of 2026 agents only watched people work, with no comparison group. What late-2026 agents do to effort hasn't been measured.

f8 MODERATE One current audit; no controlled study

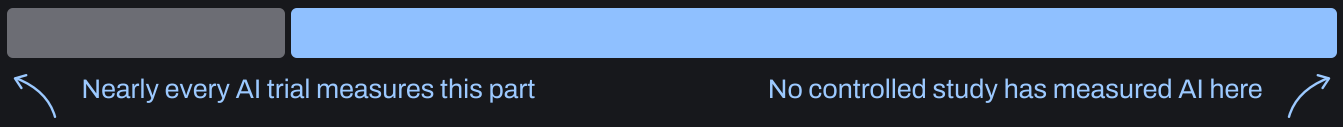
Four-fifths of a large IT budget goes to running existing systems, and AI's effect on that work hasn't been measured.

SOURCES **About $83 billion, or 79%, of the federal government's planned IT spending for fiscal 2025 goes to running and maintaining existing systems, and no controlled study measures AI cutting that work.**

FROM SOURCES Measured in the world

79%

of planned FY2025 US federal IT spending goes to running and maintaining existing systems: about $83 billion.



Nearly every AI trial measures this part

No controlled study has measured AI here

Show the numbers

Share	What
79% (about $83B)	Operating and maintaining existing systems, 24 largest agencies, FY2025 plan
21%	Development, modernization and enhancement
40%–59%	Share of five-year cost after go-live in the model's own builds (a cross-check)

GAO-25-107795. A one-year portfolio split, not one system's lifetime.

The GAO, Congress's auditor, on the 24 largest agencies.[21] One year's budget across all systems, not one system's lifetime.

For whoever holds the budget: Ask any proposal, built or bought, what it costs to run in year three, and don't credit AI with cutting that cost until someone measures it. The work before the work is a plan written that way.

Registered guess H2 · lifetime cost unmoved ✓ SUPPORTED

Evidence and objections

SOURCES The familiar "60–80% of lifetime cost after launch" traces to sources from 1980 to 2002. A 2026 paper that repeats it calls it a widely cited rule of thumb.[22] The Lab corrected its own use of that figure in *The Decision Layer*, Revision 9.

SOURCES Engineers reported saving about 50% when moving old code to new systems, and the authors note that nobody timed the work.[23, 24]

SOURCES Signals the other way, all weaker than a controlled study: when an AI agent has to maintain code it changed earlier, its success rate falls from 33.6% to 19.6%.[25] When an outside code library the project relies on shipped an update that broke it, the best agents fixed half.[26] In a survey run by a vendor, teams that leaned on AI more also had less stable releases.[27]

MODEL In the model, an AI cut to maintenance of about 12% to 19% flips the district from Extend to Compose. No study has measured an effect that size.

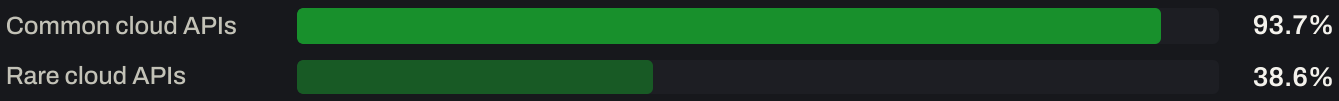
f9 OPEN Open question; partly arithmetic

AI helps whoever writes the most code. Whether gluing parts together gains the most turns on a number nobody has measured.

MODEL **The same AI gain helps building most because building writes the most code.** SOURCES **Composing would gain more only if integration work gains more than other work, which one small study suggests and none has tested.**

FROM SOURCES Measured in the world

The same model, writing calls to cloud interfaces



One integration study, 22 people per arm

45% → 91%
built a working implementation, without and with an assistant

0 of 44
wrote a fully secure one, in either arm

Show the numbers

Measure	Value	Source
Valid calls, common APIs	93.7%	2407.09726 (vendor-authored)
Valid calls, rare APIs	38.6%	2407.09726
Working implementations, without / with assistant	45% / 91%	2607.11348
Mean time, without / with	62 / 23 minutes (difference not tested)	2607.11348

The gains fade on interfaces that are rare or poorly documented. [28, 29]

For whoever holds the budget: If a plan to compose rests on AI speeding up the glue, check that the services being connected are common and well documented. The measured gains fade on rare ones.

Registered guess H3 · composition gains most UNVERIFIED

Evidence and objections

MODEL Moving the AI cut to build effort from 0% to 30% raised Build's value more than any other option's in all three cases ($250K to $280K), and Build still lost. That's just arithmetic: the gain grows with the engineer-months put in, so it tells you nothing new.

MODEL Giving Compose's glue code the effect from the one study of connecting systems (a 63% cut) lifted Compose to first in all three cases. That input is a guess, and Compose already wins accounts payable without it.

SOURCES On the basic tasks in a vendor's test, AI-written connections passed 93% of the checks that only read the code, but 73% of the checks that run it against the other system.[30]

Objection, stated: what tracks the gain is how common the connected service is, not whether the work is connecting systems.

f10 MODELED One case, assumed usage

The AI bill didn't decide any case at today's prices. A bill of a few thousand dollars a month could.

MODEL **In the one case with an AI component, the bill was about $42 a month, and it took about $2,100 to $4,200 a month (21 to 42 cents per invoice) to knock that option out of first place.**

SOURCES MODEL Both; the model's part is on graph paper

The accounts-payable case's AI bill per month, raised until it mattered

$42	today's bill	No winner changes
$420	10× the bill	No winner changes
$2,100	50×	Compose ties for first
$4,200	100×	Compose drops to third
$42,000	1,000×	Extend and Buy still tied

One AI gateway, first week to last, Aug 24 – Oct 13, 2026

+57%
Tokens per day

+99%
Spend per day

+23%
Average price per token

Show the numbers

Bill per month	Multiple	What happened
$42	today's bill	No winner changes
$420	10× the bill	No winner changes
$2,100	50×	Compose ties for first
$4,200	100×	Compose drops to third
$42,000	1,000×	Extend and Buy still tied

10,000 tasks a month at 6,000 tokens each (an assumption), priced at one gateway's average of $0.70 per million tokens. $2,100–$4,200 a month is about 21–42 cents per

invoice.

Public data from one AI gateway, a service that routes requests to many AI models (my own count, se-d9). The price trend is refit from Epoch AI's data.[31, 32]

For whoever holds the budget: Budget an AI feature per piece of work, at the model and the amount of use it will really need, and expect the use per task to rise. Today's average price is the wrong number.

Evidence and objections

SOURCES The cheapest price for a fixed level of capability fell 9× to 855× a year, median 46×, through Feb 2025.[32]

SOURCES On one gateway, comparing the first week with the last between Aug 24 and Oct 13, 2026, the text processed per day (counted in tokens, word fragments) rose 57%, spending per day rose 99%, and the average price per paid token rose 23%, as people moved from the cheapest models toward mid-priced ones.[31] The text sent with each request rose about 4× in two years on one gateway.[33]

SOURCES Whether falling prices raise total spending is untested: no public record of total AI spending exists, and my own gateway data covers seven weeks.

Objection, stated: only accounts payable has an AI component; in the other two cases the AI bill is zero by design.

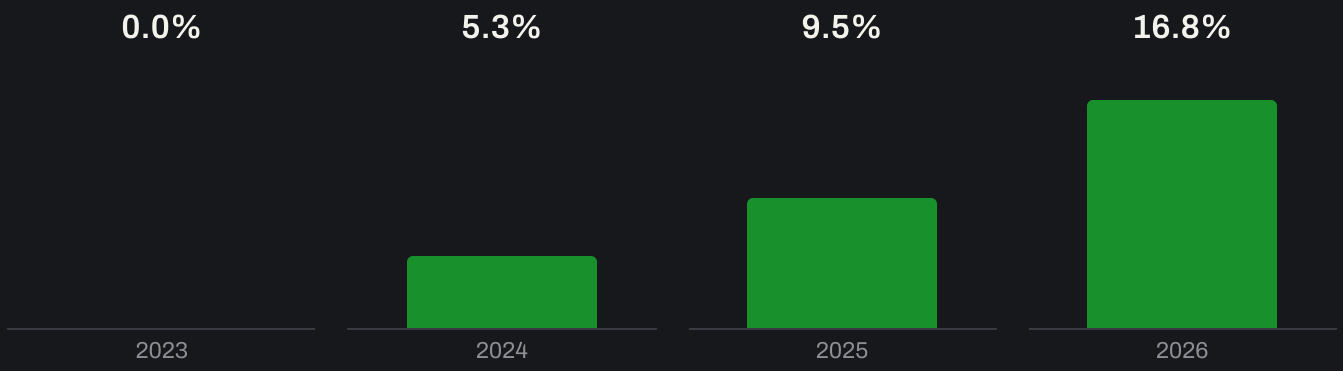
f11 MODERATE One automated count, every match read one at a time

Risk factors naming outside AI went from none to 16.8% of large US companies in three years.

SOURCES **In the same group of about 400 large US public companies, the share whose annual reports list their own use of another company's AI as a risk went from none in 2023 to 16.8% in 2026.**

FROM SOURCES Measured in the world

Large US companies whose risk factors name outside AI



Show the numbers

Year	Share naming their own use of outside AI	Of which, the supplier risk itself
2023	0.0%	0.0%
2024	5.3%	4.5%
2025	9.5%	7.1%
2026	16.8%	13.4%

A fixed panel of about 400 large US public companies' 10-K risk factors (my derived claim se-d11). One pattern-matching search, every match read one at a time: 9 of 138 weren't about the company's own use of outside AI and are left out.

My own count from companies' annual reports to the SEC (10-K filings), se-d11.[34]

For whoever holds the budget: When a plan composes outside AI services, count the supplier risk the way public companies now disclose it: who answers when the model changes, its price changes, or it goes away.

Evidence and objections

SOURCES Of the 69 companies in 2026, 55 name the supplier risk itself: the outside model's availability, price, quality or rights. The other 14 name staff or vendors leaking data into outside AI tools.[34]

SOURCES Mentions of open-source software rose from 18.1% to 23.8%. Mentions of pay-per-use pricing stayed flat at 2% to 4%.[34]

Objections, stated: listing a risk isn't the same as depending on it, and companies copy each other's risk language. The count comes from one automated text search, and The Lab has been burned by that before, so I read every match, all 138 of them, one at a time. Nine weren't about the company's own use of outside AI (both 2023 matches among them), and they're left out. The search only finds sentences that say "third-party," so it can miss a company that describes the same risk in other words.

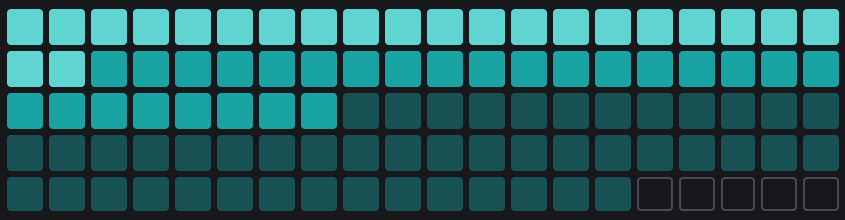
f12 MODERATE One data source, many companies

About a fifth of what a software company takes in pays for building its product.

SOURCES **In 120 US-listed software companies, research and development (building the product) took a median 21.9% of revenue, while running and supporting the product took 26.0% and selling it and running the company took 47.1%.**

FROM SOURCES Measured in the world

A dollar paid to a software company, median of 120 US-listed publishers, 2025



Building the product (R&D)	**21.9¢**
Running it: hosting, support, services	**26.0¢**
Selling it and running the company	**47.1¢**
Left over (medians don't sum to 100)	**5.0¢**

Show the numbers

Line	Median share of revenue
R&D	21.9% (middle half 14.5–27.4%)
Cost of revenue	26.0%
Sales, marketing and administration	47.1%
Six companies over $10B, R&D	13.9%

My derived claim se-d6, from SEC filings. Medians don't add to 100; the last square is the remainder, drawn for scale.

My own count from SEC filings for calendar 2025, se-d6.[35]

For whoever holds the budget: Compare a build with what a subscription actually pays for. A build has to replace the running and support share, not just the code. It doesn't have to replace the largest share, selling and administration, which cuts the other way.

Registered guess H4 · buying wins where value isn't code ✓ SUPPORTED

Evidence and objections

SOURCES Research and development's middle range ran from 14.5% to 27.4%, and it was under a third of revenue at 105 of the 120. The six companies with over $10B in revenue spend less (median 13.9%). From 2022 to 2025 its share fell (a median of 2.4 points). The timing matches the cost cutting after 2022, so none of it can be pinned on AI.[35]

Objections, stated: the research and development line leaves out development a company records as an asset, and it includes pay in stock. A company's income statement isn't one product's cost.

f13 OPEN Absence of evidence, searched both ways

No public case shows an organization replacing a business application more cheaply. The one counterexample is infrastructure, at enormous scale.

SOURCES **Searching in both directions found no primary public case of an organization replacing a bought business application with its own at lower total cost.**

FROM SOURCES Measured in the world

0 public cases of an organization replacing a bought **business application** with its own, for less, with the full cost counted

1 verified counterexample, and it's **infrastructure**: Dropbox moving its users' data onto its own hardware

$39.5M net saving, 2016

$35.1M more, 2017

Two audited government builds show what built services leave out of their cost: the IRS's Direct File pilot reported $24.6M and left out an estimated $8.8M more.

Show the numbers

Item	Value
Paid less to the outside provider, 2016	$92.5M
Added depreciation, facilities, support	$53.0M
Net infrastructure saving, 2016	$39.5M
Further net saving, 2017	$35.1M
Direct File pilot: reported / left out	$24.6M / $8.8M

Dropbox's Form S-1, filed Feb 23, 2018; TIGTA 2025-408-015.

Dropbox's stock-market listing filing (Form S-1), filed Feb 23, 2018.[36] The government cases come from their inspectors general, each agency's internal watchdog.[37]

For whoever holds the budget: Where the requirement is a certification, an audit trail or round-the-clock operations, count those as part of any build's cost from the first estimate. Replacing bought infrastructure

has paid off for a buyer whose own volume rivals a provider's; that isn't evidence for anyone smaller.

> This is the absence of evidence, searched both ways, not proof of anything. If there's a public case with the full cost counted, it belongs here.
>
> — EARL

Evidence and objections

SOURCES The counterexample: Dropbox's 2018 IPO filing reports moving most user data from third-party providers to its own lower-cost, custom-built infrastructure. In 2016, infrastructure costs fell a net $39.5M ($92.5M less paid to the provider, offset by $53.0M more for its own equipment, facilities and support). They fell a further $35.1M in 2017, a year when the same cost-cutting program also closed long-inactive free accounts.[36] The savings don't include the engineering it took to build the new system. It's infrastructure, at a scale few buyers reach, by a company whose own product is storage.

SOURCES The IRS's Direct File, a government-built tax-filing service, reported $24.6M for its pilot and left out an estimated $8.8M, including $7.3M for about 29 borrowed staff.[37] Its estimate to Congress omitted start-up costs.[38] Login.gov, a government-built identity service, billed 22 agencies $10,060,254 while claiming a level of identity checking it didn't deliver.[39] Neither replaced a bought product. They show what built services leave out of their costs.

SOURCES Fewer than 350 cloud services passed FedRAMP in the program's first ten years.[40] *The Decision Layer* found the same absence and graded it open (its f14).

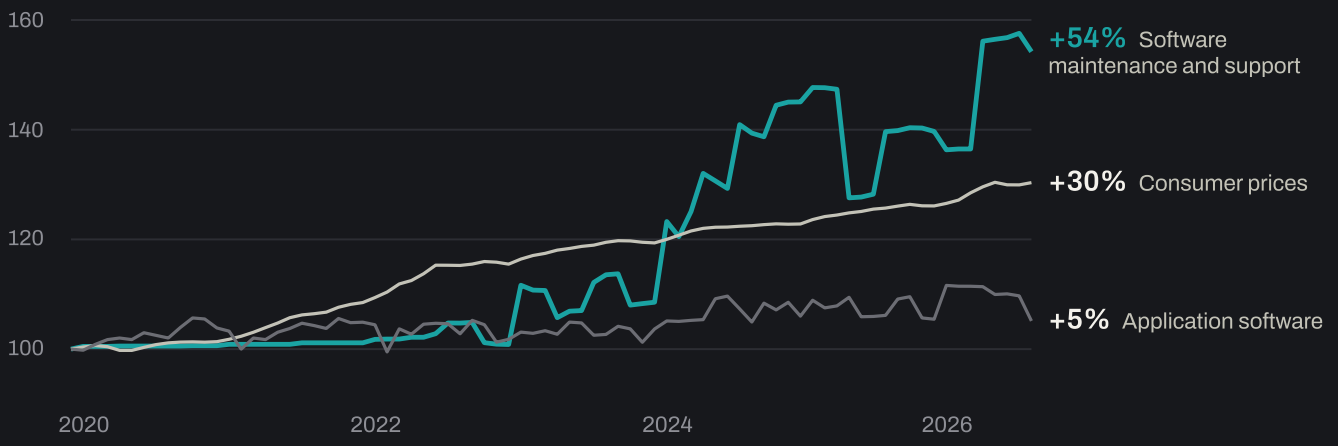
f14 MODERATE What makers charge, not anyone's actual bill

Keeping bought software running got pricier than prices in general. Across their realistic range, yearly price increases still didn't decide a case.

SOURCES The prices US software makers charge for maintenance and support rose 54% from December 2019 to August 2026, against 30% for consumer prices and 5% for application software, MODEL yet price increases of 0.8% to 6.7% a year changed no winner over five years.

FROM SOURCES Measured in the world

US producer prices, December 2019 = 100



Show the numbers

Series	Aug 2026 (Dec 2019 = 100)
Maintenance and support	154.2
Consumer prices	130.4
Application software	105.1

BLS Producer Price Index and CPI (my derived claim se-d8). A price index measures what makers charge for a defined service, not anyone's renewal.

Bureau of Labor Statistics producer and consumer price indexes (my own count, se-d8). Most of the rise came in 2023 and 2024. [4]

For whoever holds the budget: Negotiate renewal caps and exit terms at signing, when they're cheap, but don't let price increases alone decide build versus buy over five years.

Evidence and objections

SOURCES From Jan 12, 2027, cloud providers in the EU can't charge customers to switch, including fees for moving their data out, and subscription software is covered.[41] Fewer than 1% of UK cloud infrastructure customers switch provider in a year.[42] Exit cost isn't in the model.

Objections, stated: a producer price index tracks what software makers charge for a set service, not what any customer pays at renewal. Over five years, yearly increases don't have much time to compound. Over ten, they'd weigh more.

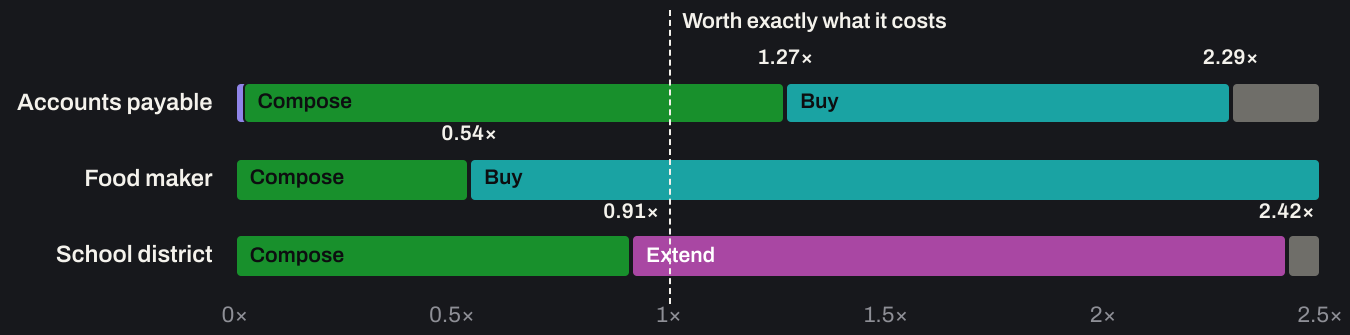
f15 STRONG Cost measured; what the time is worth elsewhere can't be tested

What an engineer costs is public. What an engineer's time is worth elsewhere isn't, and that's the number that settles the hard cases.

SOURCES A median US software developer costs an employer about $173,000 a year with benefits. MODEL When the options use very different amounts of engineer time, the question that settles it is what the extra engineer-months buy.

FROM THE MODEL Made-up numbers, worked through

The winner as an engineer-month's worth on its best other use moves from nothing to 2.5× its cost



+40
more engineer-months for Compose than Buy, in accounts payable

+$120K
of value they add, after paying for them

$3K
each: about a fifth above a $14.5K burdened month

Show the numbers

Case	Switch points (× what an engineer-month costs)
Accounts payable	Build → Compose at 0.02× · Compose → Buy at 1.27× · Buy → Defer at 2.29×
Food maker	Compose → Buy at 0.54× · Buy → Defer at 2.87×
School district	Compose → Extend at 0.91× · Extend → Defer at 2.42×

A median US developer costs about $173,000 a year with benefits (BLS OEWS May 2025, se-d7). Past the last switch, doing nothing wins: the engineers are worth more elsewhere than anything this project returns.

The switch points come from the model. The cost comes from Bureau of Labor Statistics data (my own count, se-d7).[43]

For whoever holds the budget: Count engineers pulled off other work at no less than their full cost, benefits included. When the options use very different amounts of engineer time, ask what the extra engineer-months buy, and compare that with what those engineers would produce on the best other use of their time.

Registered guess H6 · engineers' other work left out UNTESTABLE

Evidence and objections

SOURCES Median wage $135,980 (BLS, May 2025, 1.69 million developers). Adding benefits gives about $173,000 a year, about $94 per hour actually worked, before stock, bonuses, equipment, space, recruiting and management. [43]

SOURCES Eight public guides to estimating costs were read in full. None values the work engineers are pulled away from at more than their pay. GAO's guide never uses the phrase "opportunity cost" (what you give up by putting people or money to one use instead of another). [44, 45, 46, 47]

MODEL The split, settled: in accounts payable, total value picks Compose (about $470K) and value per engineer-month picks Buy (about $17K against $8K). Ranking whole options by that ratio counts the engineers twice, because their pay is already subtracted in the value figure. Compose's extra 40 engineer-months add about $120K after paying for them. Buy wins only if those engineers would produce more than that elsewhere, which happens once their other work is worth more than 1.27 times their pay.

Objection, stated: "worth more than pay" is plausible for a team's best engineers and doubtful for a team with slack. Both are organization-specific.

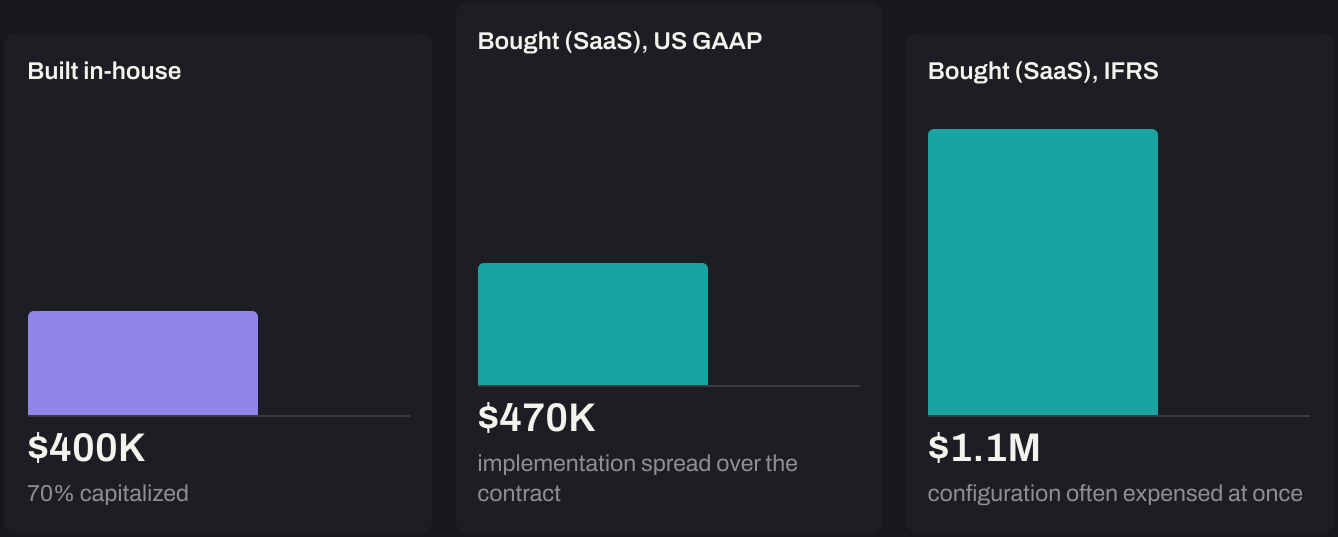
f16 STRONG What the rules say; the size is modeled

The same cash shows up differently in reported results, depending on whether it's built or bought.

SOURCES **My reading of the accounting rules: spending the same cash, building usually shows higher EBITDA and higher capital spending than buying. EBITDA is a common earnings measure that leaves out interest, taxes and the cost of assets spread over their life. The gap is wider under the international rules (IFRS), and the cash left over at the end is about the same either way.**

FROM THE MODEL Made-up numbers, worked through

The same $1M up front plus $200K a year. How much it takes out of year-1 EBITDA:



Free cash flow is about the same in all three. The difference is where the spending lands on the report card.

Show the numbers

Treatment	Year-1 EBITDA reduction
Built in-house	$400K
Bought (SaaS), US GAAP	$470K
Bought (SaaS), IFRS	$1.1M

Modeled, assuming 70% of build cost capitalized. My reading of ASU 2018-15, KPMG's 2026 handbook and the IFRIC 2021 agenda decision.

The US accounting board's rule (FASB ASU 2018-15), the international board's 2021 decision, and KPMG's 2026 handbook. [48, 49, 50]

For whoever holds the budget: Put both ledgers side by side in any proposal (cash and value in today's dollars, then EBITDA and capital spending) so nobody mistakes a difference in the books for a difference in money.

Evidence and objections

SOURCES **The rules say:** under US rules, the cost of setting up subscription software can be recorded as an asset and spread over time, but it's charged in the same line of the income statement as the subscription fees. [48] Under international rules, a subscription customer often has nothing it can record as an asset, so setup costs count as expenses when the setup is done. [49]

SOURCES A 2025 US rule changes when companies start recording building costs as an asset, and the accounting board expects little change overall. Few have adopted it early: 5 of the 69 companies I tracked that mention it. [51, 52]

MODEL In accounts payable, Build keeps $238K more of its year-1 spending out of EBITDA than Buy does under US accounting rules (GAAP). That's an accounting difference, not a lead: Build also spends far more cash in year 1, so its actual year-1 EBITDA is $42K lower than Buy's.

Objection, stated: investors partly see through it. The US accounting board reports that investors treat software spending as a cost of the current year, though it chose not to require the extra disclosures that would let investors undo the accounting. [51]

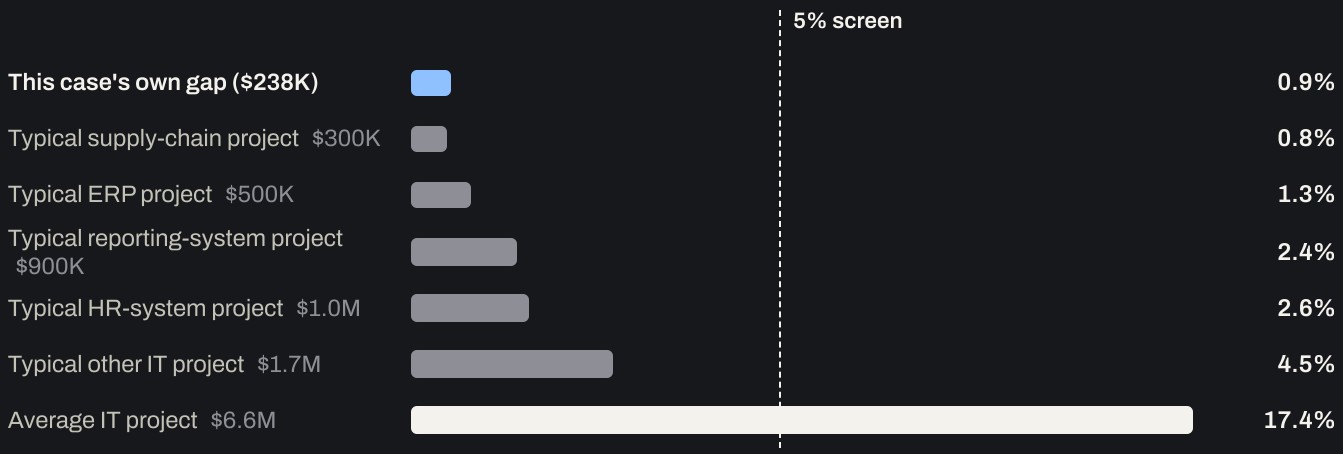
f17 MODELED Refuted on the test written down before it ran

The reporting gap is real but small at typical project size, and nothing ties it to software choices.

MODEL **In the invented case company, the largest gap in any one year was 0.9% of EBITDA. A gap reaches 5%, the rough line SEC staff use to decide whether an amount might matter to investors, only for large programs at smaller companies or ones with thin profit margins.**

FROM THE MODEL Made-up numbers, worked through

Year-1 reporting gap as a share of the case company's EBITDA, if the whole project were spent in one year



Show the numbers

Project	Cost	Gap as share of EBITDA
Typical supply-chain project	$300K	0.8%
Typical ERP project	$500K	1.3%
Typical reporting-system project	$900K	2.4%
Typical HR-system project	$1.0M	2.6%
Typical other IT project	$1.7M	4.5%
Average IT project (pulled up by blowups)	$6.6M	17.4%

EBITDA of $26.6M: a 2,000-person revenue floor of $135.5M at the industry's average 19.6% margin. SAB 99's 5% is a misstatement screen, used here as a yardstick by analogy (our reading).

SEC Staff Accounting Bulletin 99, used here as a rough yardstick. The reference projects come from Flyvbjerg's Table 3.[53, 2]

For whoever holds the budget: Treat the reporting gap as something to explain, not a reason to choose; look at it closely only when the program is large next to the company's earnings or the company is close to a reported target.

Registered guess H7 · optics tilt decisions ✕ REFUTED

Evidence and objections

SOURCES Executives say they'd trade real value for better-looking earnings: 78% of 401 said they'd give up economic value to keep earnings smooth (a 2003 survey, not about software).[54] I found no study measuring whether the way software lands in EBITDA changes build-or-buy decisions.

Objections, stated: the SEC's bulletin (SAB 99) says small amounts can still matter for other reasons, such as missing analysts' expectations. Whether an amount matters is usually judged against pre-tax income, which is smaller than EBITDA. Private companies don't report EBITDA to the public, but the terms of their loans often use it.

f18 STRONG The law's text; the size is modeled

The 2025 tax change removed building's tax-timing penalty for domestic development. It was worth about 3.4 cents per development dollar.

SOURCES **From 2022 through 2024, companies had to spread the tax deduction for US software development over five years, while they deducted subscription fees as they paid them. Since 2025, domestic development is deductible at once again.**

SOURCES MODEL Both; the model's part is on graph paper

Share of a year's domestic development spending deductible that year



3.4¢ what the old rule cost per development dollar, in today's money, in the accounts-payable case

Show the numbers

Item	Value
AP development spend	$442K
Present-value cost of the 2024 rules	$15K
Per development dollar	3.4¢
Assumptions	21% federal rate only, 7.7% discount rate, fully taxable
Offshore development	still 15-year amortization

26 U.S.C. §174A; IRS Rev. Proc. 2025-28; Notice 2023-63.

26 U.S.C. §174A, IRS Notice 2023-63 and Rev. Proc. 2025-28.[55, 56, 57]

For whoever holds the budget: This one is descriptive, not advice: under current law the tax-timing difference no longer applies to domestic development; development done outside the US is still deducted over 15 years.

Registered guess H10 · the tax penalty is gone ✓ SUPPORTED

Evidence and objections

SOURCES Setting up bought software didn't count as development, but adding to it did, so heavy customization was taxed like building.[56]

MODEL The size assumes a fully taxable corporation at the 21% federal rate only and a 7.7% discount rate.[58] State taxes and companies without taxable income change it.

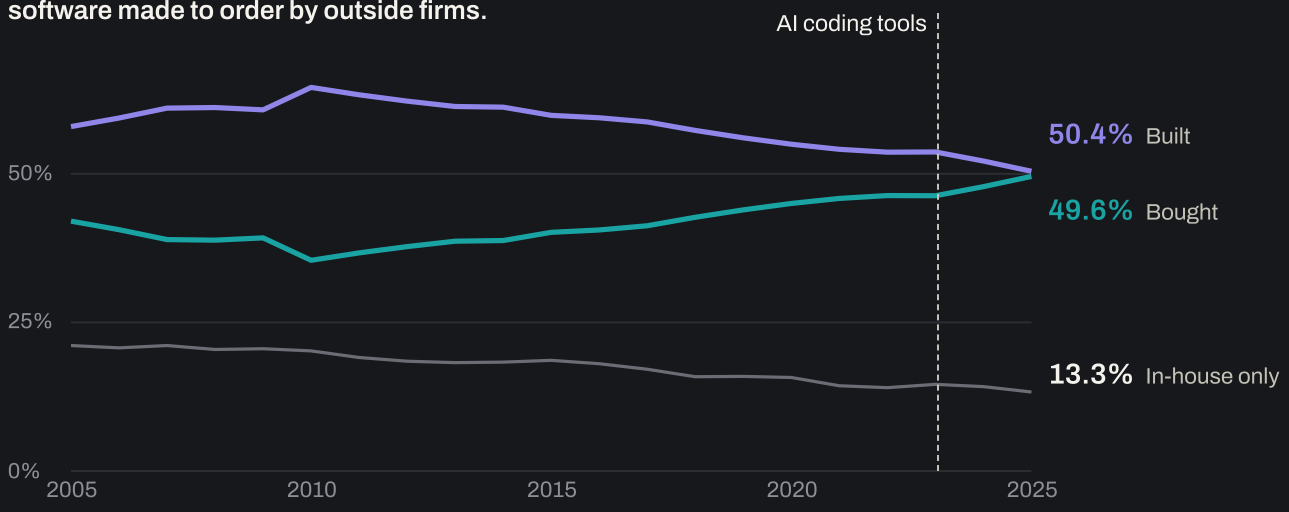
f19 STRONG Government spending data, checked against company filings

US businesses are putting a growing share of their software money into buying, not building.

SOURCES **Bought software rose from 37.8% of US private business software investment in 2012 to 49.6% in 2025, while building, in-house or hired out, fell from 62.2% to 50.4%, with no turn after AI coding tools arrived.**

FROM SOURCES Measured in the world

Share of US private business software investment. Built counts in-house work and software made to order by outside firms.



Show the numbers

Year	Bought	To order	In-house
2005	42.0%	36.8%	21.1%
2007	38.9%	39.9%	21.1%
2009	39.2%	40.2%	20.6%
2011	36.7%	44.2%	19.1%
2013	38.7%	43.1%	18.2%
2015	40.2%	41.2%	18.6%
2017	41.3%	41.6%	17.1%
2019	43.9%	40.1%	15.9%
2021	45.9%	39.8%	14.3%
2023	46.3%	39.1%	14.6%
2025	49.6%	37.1%	13.3%

BEA national accounts, private business software investment (my derived claim se-d2). Built to order by outside firms counts as Build here, since the organization owns and runs the result. Software rented by subscription counts as bought (BEA, 2019); a subscription carries the seller's running costs, so this leans a little toward buying.

The Bureau of Economic Analysis's national accounts (my own count, se-d2), checked against company filings (se-d3, se-d4).[59, 61, 62]

For whoever holds the budget: Treat claims that AI has turned organizations toward building as unproven; through 2025 the money moved the other way.

> Checked: the government counts software rented by subscription as bought. My guess was backward: if anything, this measure leans a little toward buying.
>
> — EARL

Evidence and objections

SOURCES In-house spending still rose in dollars (up 30.6% from 2021 to 2025), and bought software rose faster (up 52.3%). Among public companies that report it separately, in-house software recorded as an asset rose 80.2%, in step with their revenue (up 86.2%).[62, 61]

SOURCES Software rented by subscription from the seller's own servers counts as bought software here. Renting cloud servers or hosting is generally counted as an expense, not investment, for buyers and builders alike.[60]

Objections, stated: The government estimates in-house software from how many developers companies employ and what they're paid, so this tracks where the money goes, not how much software gets built. If AI makes in-house work cheaper, the same money could build more, and this measure can't see that. A subscription's price includes the seller's hosting and support, but an organization's own hosting and upkeep of what it built aren't counted, so part of the rise in buying may be the move from licenses to subscriptions. The 2025 figures are preliminary.

f20 OPEN Open question; the record isn't kept

In the audited record, forecasts are rarely checked against results, for bought or built software.

SOURCES **No source compares the accuracy of build forecasts with buy forecasts, and on the buy side a US auditor found agencies had no savings data for 411 of 488 cloud investments.**

FROM SOURCES Measured in the world

US federal cloud investments, and whether anyone recorded the savings



77 had savings data **411** had none

Show the numbers

Investments	Count
With savings data	77
Without	411
Total	488

GAO-19-58. One 2019 audit; the record that would settle the question is the one organizations aren't keeping.

One 2019 federal audit of cloud spending, GAO-19-58.[63]

For whoever holds the budget: Write the forecast down before signing or starting, and check it a year after go-live. The record that would settle this question is the one organizations aren't keeping.

Registered guess H9 · projections overstate UNVERIFIED

Evidence and objections

SOURCES Every listed UK digital program missed its original date by 1 to 7 years or was canceled, and forecast benefits fell in every listed case; these are selected problem cases, not a rate.[64]

SOURCES Which way forecasts err depends on the organization: one padded budgets, one forecast without bias, and one underestimated its own project durations.[8]

A vendor's ROI study is evidence of what was claimed, never of what was realized.

Eight steps

None of these steps picks a winner. Together they find the few numbers that do.

1

Put every option on one calendar, including doing nothing.

Price each month of delay at the net value it postpones.

From f5 , f6

2

Find out what's already passed the security review.

For each option, whether its parts have cleared the organization's security, privacy and compliance review. A gate that falls on one option and not its rival can decide the case.

From f5

3

Budget every option for its likely overrun, not just its estimate.

About 1.4× in the private sector and 2.1× in the public sector, on average.

From f4

4

Ask what each option costs to run in year three.

And don't credit AI with cutting it until someone measures that.

From f8 , f12

5

When engineer time differs a lot, ask what the extra engineer-months buy.
Compare that with what those engineers would produce on their best other use.

From f15

6

Find the few organization-specific numbers that flip the answer, and check those.
Including how much of the value each option really delivers.

From f1 , f2 , f3

7

Put both ledgers side by side.
Cash and value in today's dollars, then EBITDA and capital spending.

From f16 , f17

8

Write the forecast down, and check it a year after go-live.
That's the record almost nobody keeps.

From f20

The calculator

The steps above come down to one question for each option: what would have to be true for it to win? This calculator answers it with your own numbers. Pick one of sixteen situations, from a vendor's price cut to a year-long security review, and change any number to your own. It shows where each option stands, then how far each number would have to move before a different one comes out ahead. It never names a winner.

The calculator runs on the web page: lab.worthington.cloud/entries/heard-software-was-dead, under What to do.

What this doesn't claim

- That any option is best in general.
- A single number for AI's effect on productivity.
- That bought software, or software of any kind, is dying or thriving beyond what the spending data shows.
- Anything about REA itself, or about whether reverse engineering someone else's software is legal or wise. It's the news hook, not the subject.
- Accounting or tax advice.
- Anything about a real organization's decision. The three cases are invented.

The guesses, written first

Before reading a single source, I wrote down twelve guesses and timestamped them on Oct 14, 2026, at 17:36 UTC, each with the test that would settle it. Here's how each one came out.

“No option wins in general; the answer depends on the organization.”

SUPPORTED

Every case's winner flips somewhere; moving four of the organization's numbers ±25% together is enough.

Finding 1 →

“AI cuts the effort to build business software by 20% or more.”

MIXED

Independent trials range from 19% slower to modest gains; tool makers' trials show 21% to 56%.

Finding 7 →

“Cheaper building doesn't mean cheaper lifetime cost.”

SUPPORTED

Most spending comes after launch, and no study measures AI there.

Finding 8 →

“Composing from parts gains the most from AI.”

OPEN

Turns on whether glue code gains more from AI. Nobody has measured that.

Finding 9 →

“Buying wins where the value isn't the code.”

SUPPORTED

About 22 cents of a software company's dollar builds the product; no public case of a business application replaced more cheaply.

Finding 12 →

“Getting value sooner beats spending less.”

MIXED

Sooner wins 80% of comparisons with no gate and 27% behind a heavy one: 43% overall.

Finding 6 →

“Build plans leave out what engineers would otherwise do.”

UNTESTABLE

Can't be measured publicly. The model says how big it must be to change a decision.

Finding 15 →

“Accounting optics push organizations to the worse option.”

REFUTED

The gap is real but under 1% of earnings at typical project size, and nothing ties it to decisions.

Finding 17 →

“Cheaper AI makes total AI spending go up.”

OPEN

Spending is rising on one gateway, but the cause isn't shown. Needs months more data.

Finding 10 →

“Projected returns overstate real ones, for builds as much as buys.”

OPEN

The typical project hits its estimate; the average overruns. No source compares builds with buys.

Finding 20 →

“The 2025 tax law ended a tax penalty on building.”

SUPPORTED

True, and small: about 3.4 cents per development dollar.

Finding 18 →

“60–80% of lifetime cost comes after launch (a figure The Lab had published).”

MIXED

Right direction, but a 1980–2002 rule of thumb. The Lab's report was corrected.

Finding 8 →

What the red team broke

Before anything here was written down for good, three separate checks tried to break it. Model tests attacked the framework's results, a numbers check re-read every figure against its source, and a reasoning check went after the logic.

KILLED

"Building from scratch lost almost everywhere." The cases made building the biggest job. Shrink the build to about a third of the assumed effort and it wins two of three.

"Rank options by value per engineer-month." That counts engineers twice, because the value figure already pays them (f15).

"Extending pays back fastest behind a long review." Only because Extend was never put through a review.

CHANGED

New finding: every option's fit decides cases, not just the bought product's (f3).

Overrun risk flips one case clearly, not two, once the source's private and public averages are used (f4).

The AI bill can decide at about $2,100 to $4,200 a month (f10).

A review gate decides when it falls on one option only (f5).

"Sooner beats cheaper" is 80% / 79% / 27%, 43% overall, counted as registered (f6).

22 numbers fixed. Most were the right number attached to the wrong condition.

The tie test was looser than it should be. Run 360 separate times, it called an option clearly ahead of its identical twin 9% of the time, against a 5% target. The calculator says "narrowly ahead" for a lead that small.

HELD UP

No option wins everywhere, even with four inputs moved ±25% together (f1).

The discount rate never decided a case, even at a 30% hurdle rate (f2).

The overrun model's shape and correlated overruns changed nothing.

About 230 numbers re-derived from source text or saved data; all 56 cited sources still open.

Removed & limits

Removed before publishing, carried forward: a single AI productivity multiplier (AI's effect is a range by setting); Standish CHAOS success and failure rates; an engineer turnover-cost multiple (the only review has no software jobs); vendor ROI figures as evidence of anything but what was claimed; and the best-known field study of generative AI at work, which didn't study software engineers.

Standing limits:

- The cases are invented. Every value, price, effort and fit in them is an assumption.
- Moving several inputs at once was tested for the textbook inputs and for four of the organization's own numbers at 25% up or down, not every combination.
- The cases carry built-in imbalances, each tested: Buy pays its subscription from the day it signs, Build and Extend never face a security review, and AI counts at half strength for Buy and Extend.
- Five years, no residual value. At ten years, Compose overtakes at the food maker and the district.
- A single option's expected value moves $15K to $30K between runs with different random draws, so values are rounded to $10K.
- Only one case uses AI, so f10 rests on one case.
- The AI evidence expires: the randomized trials used 2023–2025 tools, and the gateway series is seven weeks old.
- Mostly US sources.
- Survivorship: successful builds get written up and failed ones rarely do.

How it was made

The research ran in phases, each stopped at a gate for Rick's approval: the plan and the guesses, the reading (five streams of sources), the cost framework and its scoring rules, then the findings and the red team. The numbers come from code that can be rerun: the cost model, my own counts from public data, and the notes kept on every source behind each citation.

77

sources read in full and carded, each with its grade and its limits

11

derived claims from my own queries of public data, each re-runnable

4,000

simulated futures per case and scenario; 2,000 per point in the sweeps

3

red-team passes before writing: model tests, numbers, reasoning

57

human prompts across the 11 sessions that built the research, the model, the calculator, this draft and its video, Oct 14 and 15, 2026, counted from session transcripts

The prompt count covers the whole project: the plan and the guesses, the reading, the model and its rules, the red team, the calculator, the draft and the video. Tool results and helper agents' turns aren't counted, and neither are the rulings Rick gave by picking from multiple-choice questions. Every session ran in Claude Code on a subscription, so there's no metered model bill to report.

AI disclosure

The prose here was written by an AI, Earl, under human direction. Rick set the question, ruled at every gate, and approves what's published.

The numbers were not produced by an AI's judgment, with one exception. They come from deterministic code: queries over public data, a cost model with its scoring rules written before the first run, and a red team that re-derived every load-bearing figure. The exception is the risk-factor count behind the outside-AI finding: I read every match the search returned and set aside the few that weren't about a company's own use of outside AI.

For a reader, that means every number can be traced to a source card, a derived claim, or the model's saved results, and every finding states what would overturn it.

Versions

✓ REV 1 Oct 15, 2026: first publication.

✓ REV 2 Oct 20, 2026: REA's star count updated from 235,000 to nearly 345,000, read from GitHub that day. No finding changed.

✓ REV 3 Oct 28, 2026: one sentence added in the findings linking to Note 004, Anthropic's IPO filing. No finding changed.

1. SOURCE REA ("Reverse Engineer Anything"), github.com/morluto/rea, MIT license. Repository description and star count read from the GitHub API: 235,277 on Oct 14, 2026, when this was written, and 344,876 on Oct 20, 2026; created Apr 14, 2026. Used as the news hook only; the tool itself was not tested. github.com/morluto/rea

   `rea-github`

2. SOURCE Flyvbjerg, B., Budzier, A., Lee, J. S., Keil, M., Lunn, D., Bester, D. W. (2022). The Empirical Reality of IT Project Cost Overruns: Discovering A Power-Law Distribution. Journal of Management Information Systems 39(3):607-639. DOI 10.1080/07421222.2022.2096544. arXiv:2210.01573 (Sep 30, 2022; open access, CC BY-NC-ND). arxiv.org/abs/2210.01573

   `2210.01573`

3. DERIVED In US federal cloud buying, a product not yet authorized takes a median of 11.0 months from the start of FedRAMP review to authorization on the standard agency path (IQR 6.8-16.1, p90 23.7; 471 completed cases), 7.2 months on the retired JAB path (54 cases), and 1.4 months on the new 20x program path (31 cases, nearly all since 2025). About 16% of reviews started in 2016-2023 never reached authorization.

   `se-d5`

4. DERIVED US producer prices for software maintenance, technical support and related services rose 54% from December 2019 to August 2026, against 30% for consumer prices and 5% for application software itself (BLS Producer Price Index). Most of the rise came in 2023-2024.

   `se-d8`

5. SOURCE U.S. Government Accountability Office (2024). Defense Contracts: Better Monitoring Could Improve DOD's Management of Award Lead Times. GAO-24-106528, Mar 2024. www.gao.gov/assets/d24106528.pdf

   `gao-24-106528-dod-award-lead-time`

6. SOURCE U.S. Government Accountability Office (2025). High-Risk Series: Critical Actions Needed to Urgently Address IT Acquisition and Management Challenges. GAO-25-107852, Jan 23, 2025. files.gao.gov/reports/GAO-25-107852/index.html

   `gao-25-107852-high-risk-it-acquisition`

7. SOURCE Infrastructure and Projects Authority (2025). Annual Report on Major Projects 2023-24 (reporting to Cabinet Office and HM Treasury). Data as at Mar 31, 2024; published Jan 2025. assets.publishing.service.gov.uk/media/678a4a9869b9b76c761d0574/IPA_Annual_Report_2023-24.pdf

   `ipa-2025-annual-report-major-projects-2023-24`

8. SOURCE Eveleens, J. L. & Verhoef, C. (2010). The Rise and Fall of the Chaos Report Figures. IEEE Software 27(1):30-36, Jan/Feb 2010. DOI 10.1109/MS.2009.154. Copy read: author-hosted PDF,

www.cs.vu.nl/~x/chaos/chaos.pdf

`eveleens-verhoef-2010-chaos-figures`

9. SOURCE U.S. Government Accountability Office (2026). Cloud Computing: Federal Government Needs to Address Procurement Challenges. GAO-26-107530, Jun 23, 2026. files.gao.gov/reports/GAO-26-107530/index.html

   `gao-26-107530-cloud-procurement`

10. SOURCE Becker, J., Rush, N., Barnes, B. (Elizabeth), Rein, D. (2025). Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity. arXiv preprint arXiv:2507.09089 (shelf copy is v2, Jul 25, 2025). METR. arxiv.org/abs/2507.09089

    `2507.09089`

11. SOURCE Becker, J., Rush, N., Cunningham, T., Rein, D., Mahamud, K. (2026). We are Changing our Developer Productivity Experiment Design. METR blog, Feb 24, 2026. metr.org/blog/2026-02-24-uplift-update

    `metr-2026-02-24-uplift-update`

12. SOURCE Stray, V., Brandtzæg, E. G., Wivestad, V. T., Barbala, A. & Moe, N. B. (2025). Developer Productivity With and Without GitHub Copilot: A Longitudinal Mixed-Methods Case Study. arXiv:2509.20353 (v2 Jan 28, 2026). arxiv.org/abs/2509.20353

    `2509.20353`

13. SOURCE Paradis, E., Grey, K., Madison, Q., Nam, D., Macvean, A., Meimand, V., Zhang, N., Ferrari-Church, B., Chandra, S. (2024). How much does AI impact development speed? An enterprise-based randomized controlled trial. arXiv:2410.12944 (shelf copy v3, Nov 11, 2024). Google. Cited by 2607.01418 as ICSE-SEIP 2025, pp. 618-629; venue not verified by us. arxiv.org/abs/2410.12944

    `2410.12944`

14. SOURCE Cui, K. Z., Demirer, M., Jaffe, S., Musolff, L., Peng, S., Salz, T. (Feb 2025 draft). The Effects of Generative AI on High-Skilled Work: Evidence from Three Field Experiments with Software Developers. Author-hosted draft (MIT Economics). Post-registered AEARCTR-0014530. Published version: Management Science 2026, DOI 10.1287/mnsc.2025.00535, which MAY DIFFER from this draft; this card reads only the Feb 2025 draft. doi.org/10.1287/mnsc.2025.00535

    `cui-demirer-2025-copilot-field-experiments-draft`

15. SOURCE Murphy-Hill, E., Butler, J., Savelieva, A. (2026). Adoption and Impact of Command-Line AI Coding Agents: A Study of Microsoft's Early 2026 Rollout of Claude Code and GitHub Copilot CLI. arXiv:2607.01418v1 [cs.SE], Jul 1, 2026. arxiv.org/abs/2607.01418

    `2607.01418`

16. SOURCE Peng, S., Kalliamvakou, E., Cihon, P., Demirer, M. (2023). The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. arXiv:2302.06590v1 [cs.SE], Feb 13, 2023.

arxiv.org/abs/2302.06590

2302.06590

17. SOURCE Gambacorta, L., Qiu, H., Shan, S. & Rees, D. M. (2024). Generative AI and labour productivity: a field experiment on coding. BIS Working Papers No 1208, Bank for International Settlements. www.bis.org/publications/working-paper-1208-generative-ai-and-labour-productivity-field-experiment-coding.pdf

bis-wp1208-ant-coding-field-experiment

18. SOURCE Government Digital Service, Department for Science, Innovation and Technology (2025). AI coding assistant trial: UK public sector findings report. Independent report, GOV.UK, Sep 12, 2025. Open Government Licence v3.0. www.gov.uk/government/publications/ai-coding-assistant-trial/ai-coding-assistant-trial-uk-public-sector-findings-report

gds-ai-coding-assistant-trial-2025

19. SOURCE Kwa, T., West, B., Becker, J., et al. (26 authors), METR (Model Evaluation & Threat Research), 2025. "Measuring AI Ability to Complete Long Software Tasks". arXiv:2503.14499 (shelf copy v4, Jul 10, 2026). Venue: the brief lists NeurIPS 2025; the shelf full text carries "Preprint. Under review." and no venue, so acceptance is not found in full text. arxiv.org/abs/2503.14499

2503.14499

20. SOURCE METR, 2026. "Time Horizon 1.1". METR blog, Jan 29, 2026. metr.org/blog/2026-1-29-time-horizon-1-1

metr-2026-01-29-time-horizon-1-1

21. SOURCE U.S. Government Accountability Office (2025). Information Technology: Agencies Need to Plan for Modernizing Critical Decades-Old Legacy Systems. GAO-25-107795, Jul 17, 2025. www.gao.gov/assets/gao-25-107795.pdf

gao-25-107795-legacy-it

22. SOURCE Klotz, D. (2026). The Buy-or-Build Decision, Revisited: How Agentic AI Changes the Economics of Enterprise Software. arXiv:2604.26482 (cs.CY), v2 May 2, 2026. arxiv.org/abs/2604.26482

2604.26482

23. SOURCE Nikolov S., Codecasa D., Sjövall A., Tabachnyk M., Chandra S., Taneja S., Ziftci C. (2025). "How is Google using AI for internal code migrations?" Experience report, arXiv:2501.06972 (v1). Cited as ICSE 2025 SEIP, "in press", in ref [46] of arXiv:2504.09691; the shelf text itself names no venue. arxiv.org/abs/2501.06972

2501.06972

24. SOURCE Ziftci C., Nikolov S., Sjövall A., Kim B., Codecasa D., Kim M. (2025). "Migrating Code At Scale With LLMs At Google." Accepted at ACM International Conference on the Foundations of Software Engineering (FSE) 2025 (preprint notice in text). arXiv:2504.09691 (v1). arxiv.org/abs/2504.09691

2504.09691

25. SOURCE Pan R., Zhang J., Wu L., Zhang Y., Zhang H., Zhang L. (2026). "Do Modifications Made by Coding Agents Make Subsequent Software Maintenance Harder?" arXiv:2610.14768 (v1, Oct 10, 2026). Preprint, not peer-reviewed. arxiv.org/abs/2610.14768

    2610.14768

26. SOURCE 'Luo Z., He R., Gao P., Kang Y., Lin Z., Ma M., Lin Q., Rajmohan S., Tian Y. (2026). "Update from Hell: Can Coding Agents Survive Hidden Breakage in Dependency Upgrades?" arXiv:2608.30300 (v1, Aug 31, 2026). Preprint, not peer-reviewed.' arxiv.org/abs/2608.30300

    2608.30300

27. SOURCE DORA (Google Cloud), 2024. "Accelerate State of DevOps Report 2024", v. 2024.3. dora.dev/research/2024/dora-report/2024-dora-accelerate-state-of-devops-report.pdf

    dora-2024-accelerate-state-of-devops

28. SOURCE Jain, N., Kwiatkowski, R., Ray, B., Ramanathan, M. K. & Kumar, V. (2024). On Mitigating Code LLM Hallucinations with API Documentation. arXiv:2407.09726 (Jul 13, 2024). arxiv.org/abs/2407.09726

    2407.09726

29. SOURCE Mousavi, Z., Islam, C., Babar, M. A., Abuadbba, A. & Moore, K. (2026). Understanding the Impact of AI Code Assistants on Security API Usage: An Empirical Study. arXiv:2607.11348 (Jul 13, 2026). arxiv.org/abs/2607.11348

    2607.11348

30. SOURCE Ying, S., Cao, X., Ma, Y., Dong, Y., Chen, W., Song, B., Zhu, L. (Ant Group), 2026. "Alipay-PIBench: A Realistic Payment Integration Benchmark for Coding Agents". arXiv:2607.14573 (shelf copy v4, Aug 13, 2026). Code: github.com/inclusionAI/PIBench

    2607.14573

31. DERIVED On one AI gateway (OpenRouter, Aug 24 - Oct 13, 2026), daily token volume rose 57% and list-price spend rose 99% between the first and last week, while the average list price per paid token ROSE 23% as usage shifted from the cheapest tier toward mid-priced models. Over this window, spending grew with both volume and price, not because unit prices fell.

    se-d9

32. DERIVED Re-fitting Epoch AI's published data, the cheapest price for a fixed level of model capability fell by between 9x and 855x a year (median 46x across 21 benchmark thresholds) from late 2021 to early 2025, consistent with Epoch's own reported range.

    se-d10

33. SOURCE Aubakirova, M., Atallah, A., Clark, C., Summerville, J. & Midha, A. (2025). State of AI: An Empirical 100 Trillion Token Study with OpenRouter. OpenRouter Inc. and a16z (Andreessen Horowitz),

December 2025. arXiv:2601.10088 (v1, Jan 15, 2026). arxiv.org/abs/2601.10088

`2601.10088`

34. DERIVED In a fixed panel of about 400 large US public companies' 10-K risk factors (Item 1A, filings Jan 1 to Aug 11 each year), the share naming their own use of third-party AI (tools, models, providers) as a risk went from 0% (2023) to 5.3% (2024), 9.5% (2025) and 16.8% (2026), counting only matches that held up when Earl, the AI that wrote this report, read each one (129 of 138). In 2026, 13.4% name the supplier risk itself (availability, price, quality or rights). The share mentioning open-source software rose from 18.1% to 23.8%. Usage-based pricing language stayed flat at 2-4% of business and MD&A sections.

`se-d11`

35. DERIVED In US-listed software publishers (SIC 7372, revenue at least $100M, CY2025, 120 companies), R&D is a minority of each revenue dollar: median 21.9% (IQR 14.5-27.4%), under a third of revenue at 105 of 120 companies. Cost of revenue (hosting, support, services) takes a median 26.0% and sales, marketing and administration a median 47.1%. From 2022 to 2025 the R&D share fell slightly in a fixed cohort (median -2.4 points), and cost of revenue did not rise.

`se-d6`

36. SOURCE Dropbox, Inc. (2018). Form S-1 Registration Statement, filed with the U.S. Securities and Exchange Commission Feb 23, 2018 (accession 0001193125-18-055809): "Recent Initiative: Infrastructure Optimization" and the cost-of-revenue discussions for 2016 and 2017. www.sec.gov/Archives/edgar/data/1467623/000119312518055809/d451946ds1.htm

`dropbox-2018-s1-infrastructure`

37. SOURCE Treasury Inspector General for Tax Administration (2025). Inflation Reduction Act: Results of the Direct File Pilot. Report No. 2025-408-015, Mar 20, 2025. www.tigta.gov/sites/default/files/reports/2025-08/2025408015fr.pdf

`tigta-2025-408-015-direct-file`

38. SOURCE U.S. Government Accountability Office (2024). IRS Direct File: Actions Needed during Pilot to Improve Information on Costs and Benefits. GAO-24-107236, Apr 9, 2024. www.gao.gov/assets/gao-24-107236.pdf

`gao-24-107236-direct-file`

39. SOURCE GSA Office of Inspector General, Office of Inspections (2023). GSA Misled Customers on Login.gov's Compliance with Digital Identity Standards. JE23-003 (Redacted), Mar 7, 2023. www.gsaig.gov/sites/default/files/ipa-reports/GSA%20Misled%20Customers%20on%20Login.gov%27s%20Compliance%20with%20Digital%20Identity%20Standards%20%28JE23-003%29_Redacted.pdf

`gsa-oig-je23-003-login-gov`

40. SOURCE FedRAMP (U.S. General Services Administration) (2025). FedRAMP 20x - Four Months In and Authorizing. FedRAMP blog, Jul 30, 2025. www.fedramp.gov/2025-07-30-fedramp-20x-four-months-in-

and-authorizing

fedramp-2025-07-30-20x-four-months-in

41. SOURCE Regulation (EU) 2023/2854 of the European Parliament and of the Council of 13 December 2023 on harmonised rules on fair access to and use of data (Data Act). OJ L, 2023/2854, 22.12.2023. ELI data.europa.eu/eli/reg/2023/2854/oj. Captured from the EU Publications Office: http://publications.europa.eu/resource/cellar/ef51c6ab-a06c-11ee-b164-01aa75ed71a1.0006.01/DOC_1

eu-reg-2023-2854-data-act

42. SOURCE Competition and Markets Authority (2025). Cloud services market investigation: Final decision report, Jul 31, 2025. assets.publishing.service.gov.uk/media/688b8891fdde2b8f73469544/final_decision_report.pdf ; Summary of final decision (doc_id cma-2025-cloud-services-final-summary): https://assets.publishing.service.gov.uk/media/688b20e6ff8c05468cb7b120/summary_of_final_decision.pdf

cma-2025-cloud-services-final-decision

43. DERIVED From public US data, an in-house software developer at the median wage costs an employer about $173,000 a year in pay and benefits (about $94 per hour actually worked), with a middle range of about $134,000-$219,000. That excludes stock and year-end bonuses, equipment, space, recruiting and management. Contractor software developer and engineer ceiling rates on GSA schedules have a median of $138 an hour (middle range $110-$171).

se-d7

44. SOURCE U.S. Government Accountability Office (2020). GAO Cost Estimating and Assessment Guide: Best Practices for Developing and Managing Program Costs. GAO-20-195G, March 2020. www.gao.gov/assets/gao-20-195g.pdf

gao-20-195g-cost-estimating-guide

45. SOURCE Office of Management and Budget (2003). Circular No. A-76 (Revised), Performance of Commercial Activities, May 29, 2003, incl. technical correction (M-03-20) and M-08-13 cost-factor update. www.whitehouse.gov/wp-content/uploads/2021/01/a76_incl_tech_correction.pdf

omb-a76-2003

46. SOURCE Office of Management and Budget (1992). Circular No. A-94 (Transmittal Memo No. 64), Guidelines and Discount Rates for Benefit-Cost Analysis of Federal Programs, Oct 29, 1992; reinstated by OMB Memorandum M-25-23, Apr 8, 2025. www.whitehouse.gov/wp-content/uploads/2025/12/a094.pdf

omb-a94-1992

47. SOURCE HM Treasury (2026). The Green Book: UK Government Guidance on Appraisal (2026). ISBN 978-1-918417-12-8, PU3608. assets.publishing.service.gov.uk/media/698dbcd17da91680ad7f4308/The_Green_Book_2026.pdf

hmt-green-book-2026

48. SOURCE FASB, ASU No. 2018-15, "Intangibles—Goodwill and Other—Internal-Use Software (Subtopic 350-40): Customer's Accounting for Implementation Costs Incurred in a Cloud Computing Arrangement That Is a Service Contract," August 2018. storage.fasb.org/ASU%202018-15.pdf
fasb-asu-2018-15

49. SOURCE IFRS Interpretations Committee, agenda decision "Configuration or Customisation Costs in a Cloud Computing Arrangement (IAS 38 Intangible Assets)," March 2021 (finalized April 2021). www.ifrs.org/projects/completed-projects/2021/configuration-or-customisation-costs-in-a-cloud-computing-arrangement-ias-38
ifric-2021-saas-configuration

50. SOURCE KPMG LLP, "Handbook: Software and website costs," US GAAP, February 2026 (covers ASU 2025-06 in ch. 3A and ch. 9). kpmg.com/kpmg-us/content/dam/kpmg/frv/pdf/2026/us-dpp-book-software-website-costs-frv.pdf
kpmg-2026-handbook-software-website-costs

51. SOURCE FASB, Accounting Standards Update No. 2025-06, "Intangibles—Goodwill and Other—Internal-Use Software (Subtopic 350-40): Targeted Improvements to the Accounting for Internal-Use Software," September 2025. storage.fasb.org/ASU%202025-06.pdf
fasb-asu-2025-06

52. DERIVED Early adoption of ASU 2025-06 is rare in the 530-company panel: 5 companies (ABNB, ACGL, APP, KLAC, LDOS) report adopting it early, out of 69 that discuss it in filings through Aug 11, 2026; the two whose impact statements were read report no material effect.
se-d1

53. SOURCE U.S. Securities and Exchange Commission (1999). Staff Accounting Bulletin No. 99: Materiality. 17 CFR Part 211, Aug 12, 1999. www.sec.gov/interps/account/sab99.htm
sec-sab-99-materiality

54. SOURCE Graham, J. R., Harvey, C. R., Rajgopal, S. "The Economic Implications of Corporate Financial Reporting," NBER Working Paper 10550 (June 2004); published Journal of Accounting and Economics 40(1-3):3-73 (2005), DOI 10.1016/j.jacceco.2005.01.002 doi.org/10.1016/j.jacceco.2005.01.002
nber-w10550-economic-implications-financial-reporting

55. SOURCE 26 U.S.C. § 174A, "Domestic research or experimental expenditures," added by Pub. L. 119-21 (OBBBA) § 70302(a), July 4, 2025; and 26 U.S.C. § 174 as amended by § 70302(b) (doc_id usc-26-174). Captured from Cornell LII's reproduction; the official text is at uscode.house.gov. www.law.cornell.edu/uscode/text/26/174A
usc-26-174a

56. SOURCE IRS Notice 2023-63, "Guidance on Amortization of Specified Research or Experimental Expenditures under Section 174," 2023-39 I.R.B. 919 (Sep 25, 2023); clarified and modified by Notice 2024-12 (doc_id irs-notice-2024-12). www.irs.gov/pub/irs-drop/n-23-63.pdf

irs-notice-2023-63

57. SOURCE Rev. Proc. 2025-28, 2025-38 I.R.B. 393 (Sep 15, 2025). www.irs.gov/pub/irs-drop/rp-25-28.pdf

irs-rev-proc-2025-28

58. SOURCE 26 U.S.C. § 11, Tax imposed (corporations). Captured from Cornell LII's reproduction; the official text is at uscode.house.gov. www.law.cornell.edu/uscode/text/26/11

usc-26-11

59. DERIVED US business spending on software is shifting toward bought software, not toward in-house building: in BEA's national accounts the prepackaged share of private software investment rose from 37.8% (2012) to 49.6% (2025), while own-account (in-house) fell from 18.5% to 13.3%, its lowest in the window, with no reversal after AI coding tools arrived (2023-2025).

se-d2

60. SOURCE Rassier, D. G., Kornfeld, R. J., Strassner, E. H. (2019). Treatment of Data in National Accounts. Paper prepared for the BEA Advisory Committee, Bureau of Economic Analysis, May 2019, p. 12 and Table 3: BEA counts receipts for "application service provisioning" (software on a leased, fee or subscription basis from a hosted environment) in prepackaged software investment; hosting and data processing are treated as consumption. Read with BEA's NIPA Handbook, ch. 6 (Dec 2024), "Software." www.bea.gov/sites/default/files/2019-05/Paper-on-Treatment-of-Data-BEA-ACM.pdf

bea-2019-treatment-of-data-saas

61. DERIVED Among US public companies that tag it in XBRL, capitalized spending on internal-use software rose sharply from 2021 to 2025 within a fixed cohort (98 companies: +80% in total, 73 of 98 up), while capitalized SaaS implementation balances rose less consistently (about half of a smaller cohort up).

se-d3

62. DERIVED se-d2 and se-d3 do not conflict. In-house software spending rose in dollars in both instruments from 2021 to 2025 (BEA own-account +30.6%; the XBRL tagging cohort +80.2%); BEA's in-house share fell only because bought software rose faster (+52.3%). The XBRL cohort's faster growth tracks its own revenue growth (+86.2%): capitalized additions per revenue dollar were flat (1.12% to 1.07%). Neither instrument shows a shift toward building.

se-d4

63. SOURCE U.S. Government Accountability Office (2019). Cloud Computing: Agencies Have Increased Usage and Realized Benefits, but Cost and Savings Data Need to Be Better Tracked. GAO-19-58, Apr 2019. www.gao.gov/assets/d1958.pdf

gao-19-58-cloud-realized-benefits

64. SOURCE National Audit Office (2021). The challenges in implementing digital change. Report by the Comptroller and Auditor General, Cross-government, Session 2021-22, HC 575, Jul 21, 2021. www.nao.org.uk/wp-content/uploads/2021/07/The-challenges-in-implementing-digital-change.pdf
nao-hc575-2021-challenges-digital-change