Assette

Yes, AI Can Build Your Pitchbooks and Client Reports.

www.assette.com



Assette

AN ASSETTE PERSPECTIVE

Yes, AI Can Build Your Pitchbooks and Client Reports.

The question is what it costs to own them for the next forty quarters.

THE QUESTION

The question senior management is asking:

Someone at your firm has seen the demonstration. New data files and last quarter's pitchbook go in; this quarter's version comes out, figures and charts updated. It takes minutes. The obvious question follows: do we still need vendor software to produce sales and client content?

It is a fair question, and it deserves better than a vendor's reflex. We have run both versions of the experiment ourselves — using AI directly to produce presentations, and using AI throughout our own engineering work. The short answer is that the demonstration is real. The longer answer is that it splits into two very different questions, with two very different answers, and the industry has already lived through the second one once before.

Two kinds of systems

Everything downstream depends on one distinction.

Deterministic

The same input produces the same output, every time. Traditional software, calculation engines, and rules-based disclosure logic work this way. Deterministic systems have bugs — but a bug is reproducible. It fails the same way every time, so it can be found, diagnosed, fixed once, and it stays fixed. Errors converge toward zero over time.

Probabilistic

The same input can produce different outputs. Large language models generate each response fresh. A wrong number this quarter is not caused by a flaw that can be patched — it is inherent to how the system works. Errors do not converge. Every run is a new draw, and the fortieth quarter carries the same risk as the first.

The history of computing is a long argument for why the first property matters. In 1994, Intel's Pentium processor returned wrong answers on certain division operations. Replacing the flawed chips cost Intel a $475 million pre-tax charge, by Intel's own accounting.[1] The error was reproducible — same inputs, same wrong answer, every time — so it could be characterized and permanently corrected. Nobody has worried about processor arithmetic since. Early compilers had bugs too; each, once found, stayed fixed — which is why aviation and medical software is compiled today without a second thought.

Deterministic errors are debts you pay off. Probabilistic errors are rent you pay forever.

The demonstration works. The process does not.

Point AI at last quarter's documents and it will produce this quarter's. We have done it. What breaks is not the mechanics but the meaning of "accurate" at production scale.

A single client report can carry hundreds of individual figures across tables, charts, and footnotes. Even at 99.5% per-figure accuracy — generous for data transcription — expect one or two errors per document, and expect not to know which ones. In regulated client communications, a wrong performance figure is not a typo. It is a compliance incident.

The instinct that work would shift from production to review is correct, and it understates the problem. Verifying a document you did not build is harder than building it: to confirm a figure, the reviewer must independently recompute it, which requires the same data access and calculation logic the production process needed. And these errors are unusually difficult to catch. Deterministic systems fail loudly — a broken formula, a blank cell. A language model fails quietly, producing a wrong number that looks exactly like a right one.

Then there is who does the reviewing. The people qualified to verify a composite return, a benchmark pairing, or a disclosure selection are the same experienced operations and compliance professionals the automation was meant to free. This path does not redeploy them. It converts them into full-time proofreaders of machine output — a harder and less satisfying job than the one they had.

Disclosures make it concrete. Disclosure selection is rules-based: jurisdiction, vehicle, performance type, reporting period. A rules engine guarantees the right disclosure appears. AI predicts it. Compliance needs a guarantee.

The natural response is a proof of concept — two or three simple reports, low stakes. It will very likely succeed, and that is what misleads: PoCs are small by design — hand-picked templates, someone checking every figure. The economics invert when review must cover hundreds of documents nobody hand-picked, every quarter, indefinitely. A successful PoC proves the demonstration, not the process.

The industry has run this experiment before

If AI cannot reliably produce the documents, can it write the software that does? That is the second question, and it is much older than AI.

In the 1990s, a wave of tools put software creation within reach of business users and small teams. Visual Basic, Access, and Excel macros meant a capable analyst could build a real application in a few weeks — and across the investment industry, thousands did. Performance calculators, client report generators, fee engines, attribution tools. The creation problem was genuinely solved.

What was never solved was everything after creation. The tools worked; the ownership did not. Those applications were undocumented, understood by one person, dependent on custom data formats and impossible to audit. Every time a source changed its format, the program broke and had to be reworked. When the analyst left, the logic left with them. Most of that generation of internal software was eventually replaced by vendors — and the spreadsheet sprawl that content automation platforms displace today is, in large part, the surviving residue of that build cycle.

The lesson was specific: the constraint on internal software was never the difficulty of writing it. It was the cost of owning it. AI is that moment again, with far better tools — aimed at the same part of the problem.

The ownership iceberg

Writing code is roughly 20–30% of the lifetime cost of a system like this. AI has genuinely collapsed that portion. It has done far less to the rest.

Knowing what to build. Composite logic, client-provided template handling, GIPS-adjacent rules, benchmark licensing restrictions, share-class disclosure matrices, the specific way your firm treats a partial period. This is requirements knowledge accumulated over years, and AI cannot supply it. Unless a firm has already written these rules down — in specifications, decision logs, or code — there is nothing for the model to read. In most firms this knowledge lives in the heads of a few operations and compliance people.

Data engineering. Connections to source systems, transformations, and calculations; handling restatements, late data, and format changes when a source updates a feed. This is the part that breaks quarterly, forever.

Maintenance without an end date. Regulation does not stand still. Every rule amendment, every GIPS update, every benchmark license change becomes an internal project competing for priority against everything else the firm needs built. The build decision is not one decision — it is a standing commitment to re-decide "is this worth engineering time" every time the rules move.

Put five years of that into a model and the shape is consistent whatever your numbers: a build in the low hundreds of thousands, one to one-and-a-half engineers forever, hosting and tooling, and AI usage costs no one can budget in advance. The total lands in the millions. The decision is not build-versus-buy on the build. It is build-versus-buy on the decade.

Firms with substantial IT organizations settled this question years ago for software of this kind: nobody builds their own CRM anymore, least of all the largest firms. Not because it could not be done, but because it was never worth owning.

The SaaS-versus-AI debate, taken seriously

There is a real argument on the other side, and it is stronger than most vendors admit.

Much enterprise software is thin: screens over a database, generic by design, and requiring an expensive configuration project to become useful to any particular firm. A large CRM implementation can cost several times the license, run for a year or more, and produce something the firm effectively specified itself. If you were paying to build your own version anyway, and AI collapses the labor of building, then building outright starts to look rational.

Klarna became the case study for this thesis, and the details are instructive. The company announced it was shutting down Salesforce and Workday.[2] But it did not replace them with AI-built software; it moved to other vendors and a blend of third-party and internal systems.[3] Its more aggressive AI substitution — replacing customer service staff — was reversed within eighteen months, with the CEO telling Bloomberg the company had gone too far, having focused on cost at the expense of quality.[4] The most-cited proof that AI replaces bought software turns out to be a story about switching vendors — and rediscovering where machines underperform.

Still, the underlying distinction holds, and it is worth stating precisely rather than defensively:

AI erodes the case for generic software that requires heavy configuration to become industry-specific. It does not erode the case for purpose-built software, where the industry logic is already accumulated and someone else carries the maintenance.

Sales and client content in investment management sits firmly in the second category. The value was never the interface. It is composite handling, disclosure rules, consultant database formats, platform and home-office requirements, client-imposed templates, and the thousand accumulated edge cases of communicating regulated performance to institutional allocators and wealth clients. That knowledge is the product. Code is how it is expressed.

AI is a multiplier

The most common error in this debate is treating AI as something that only helps the buyer build. It helps everyone, and it multiplies whatever it is applied to.

Applied to a blank page, AI produces a first version quickly. Applied to decades of data connections, compliance logic, and accumulated edge cases, it compounds — faster delivery, broader coverage, new requirements met sooner. It also compresses the part of buying that always felt most expensive: implementation. Configuration, data mapping, and template conversion are exactly the labor AI is good at, so the deployment cost that made buying feel heavy is falling for the same reason building feels light.

Both sides of the decision get the AI dividend. Only one side gets it multiplied by work already done.

Where AI genuinely creates advantage

None of the above says AI fails at this work. It says document mechanics is the wrong place to point it. There are three places where the same capability creates advantage instead of risk.

Research at a scale no team can staff.

Reading and synthesizing research inputs — filings, transcripts, sell-side work, news, alternative data — at volumes no group of analysts could work through. The value is not summarizing faster than an analyst. It is taking the investment process deeper than headcount allows: every holding, every quarter, every stated thesis tested against what has actually changed. The output is not a document. It is a question nobody had thought to ask.

Internal tools that create edge.

Applications that examine the firm's information in ways no vendor would build, because no vendor holds the firm's investment beliefs. This is the same build decision examined above, made in the one category where owning the software is the entire point: if a tool encodes how your firm sees markets, it should not be something a competitor can buy.

Storytelling worth remembering.

Commentary and investment narrative are where value sits in language rather than arithmetic. Exploring several ways to explain a drawdown; finding the analogy that makes a process land for a trustee who is not an investment professional; pressure-testing whether the story the materials tell matches the decisions actually made. This is craft, and AI is a real collaborator in it — while the judgment about what is true and what matters stays with the firm.

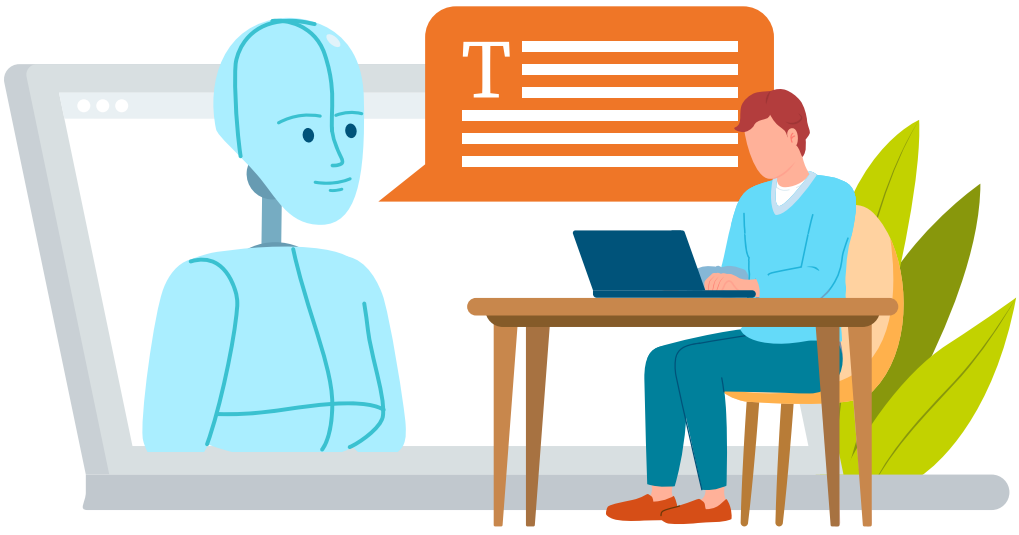
There is a pattern across all three. Probabilistic systems are powerful where variation, exploration, and language are the point, and dangerous where correct numbers are the point. Research synthesis, tool-building, and narrative craft are the first kind of work. Producing performance figures in a sales deck is the second. Most of the disappointment with AI in this industry comes from pointing it at the second kind of work and judging it by the standards of the first.

The bottom line

Pointing AI at last quarter's documents works as a demonstration and fails as a process; it moves cost from production to verification, and verification is the more expensive, riskier half. Having AI write the software genuinely works — and converts the question into build-versus-buy, where the answer is decided by the decade of ownership rather than the weeks of construction. AI changed who benefits from cheap software creation. It did not change who should own the maintenance.

The better use for scarce engineering and AI talent is the work described above, where advantage is actually available: research at a scale no team can staff, tools that encode how your firm sees markets, and narrative that makes an investment story land and stay landed.

Underneath the whole question is how business is actually won in the investment industry. Firms win and keep AUM on three things: performance, relationships, and the story that explains what drives performance — the philosophy, the process, the people. The story matters most precisely when the numbers are not in your favor. No firm fully controls performance. Every firm controls the others. The story is yours alone and deserves real effort. Own the story. Buy the machinery.



1. Intel Corporation, *1994 Annual Report* — "$475 million pre-tax charge… to cover replacement and write-off of these microprocessors." Announced January 17, 1995.
2. Sebastian Siemiatkowski, Klarna earnings call, September 2024, as reported by Seeking Alpha and subsequently widely covered.
3. Klarna statement to *CX Today*, December 2024, confirming replacement of Salesforce CRM and Workday functionality with alternative vendors and a blend of third-party and internal systems.
4. Sebastian Siemiatkowski, interview with *Bloomberg*, May 2025.