

CUSTOMER CASE STUDY

The CIO Who Built His Quant Desk in Under an Hour

How a skeptical CIO went from "statistical collation" to champion in under two hours, with a live GARCH calibration on his own strategy returns

Scottsdale RIA | July 2026 | Institutional Asset Management

EXECUTIVE SUMMARY

Udayan Mitra, CIO of a \$2.6B Scottsdale RIA with 34 advisors, had tried generalist AI: ChatGPT, Claude, Gemini, Bloomberg's "Ask B", and found it wanting.

His critique was precise: generalist AI is not intelligence. It is statistical collation of internet consensus. In a 28-minute first call, he described exactly what he wanted: a system that could calibrate a GARCH volatility model on his strategy returns, run walk-forward validation, and integrate with Black-Litterman portfolio optimization.

In a 54-minute demo thereafter, Ben McMillan did exactly that, live, on Udayan's own strategy data, in under 10 minutes. By lunch, Udayan had pivoted from skeptic to internal champion: "Let's get this in front of my advisors."

CONTENTS

1 The Problem

2 The Tool

3 The Search

4 The Results

5 The Build

6 The Bigger Picture

How a skeptical CIO went from "statistical collation" to champion in under two hours, with a live GARCH calibration on his own strategy returns

Udayan Mitra, CIO of a \$2.6B RIA with 34 advisors, had tried generalist AI: ChatGPT, Claude, Gemini, Bloomberg's "Ask B", and found it wanting. But his critique wasn't the usual "AI hallucinates." It was more precise: generalist AI is not intelligence. It is statistical collation of internet consensus. For an institutional investor whose edge depends on thinking differently from the crowd, a system that converges on the crowd's opinion is useless.

In a 28-minute first call, Udayan described exactly what he wanted: a system that could calibrate a GARCH volatility model on his strategy returns, run walk-forward validation, track in-sample vs. out-of-sample iterations over time, and integrate with Black-Litterman portfolio optimization, all while remembering his methodology, his parameter choices, and his reasoning chain.

In a 54-minute demo three weeks later, Ben McMillan did exactly that. He spun up a test agent, uploaded Udayan's actual Apex Equity Growth returns, data McMillan hadn't seen before the call, and in under 10 minutes ran a full GARCH(1,1) calibration with walk-forward validation across five model variants, autocorrelation analysis, Markov-switching regime detection, and generated formatted charts with an executive summary. The critic subagent caught desmoothing concerns on the actively managed equity returns. Udayan watched it reason in real time, then asked for the charts by email.

By lunch, Udayan had pivoted from power user to internal champion: "Let's get this in front of my advisors." The trainer who onboards 's 34 advisors is coming to the next meeting.

The real story isn't the GARCH model. It's that a skeptical, quantitatively sophisticated CIO described his ideal analytical workflow in his own words, watched it get built live on his own data, and became an internal champion, all within two meetings. No developers. No six-month vendor procurement. Just an advisor who knew what he wanted and a system that could build it.



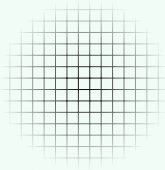
The Problem

Udayan Mitra had tried AI. As CIO of a \$2.8B Scottsdale RIA with 34 advisors, he had used ChatGPT, Claude, Gemini, and Bloomberg's "Ask B" for market research, PPM review, rolling correlations, due diligence on dubious proposals, and weekly market briefs for his advisor fleet.

He was not impressed.

"Statistical Collation, Not Intelligence"

Udayan's critique was epistemic, not operational. Generalist AI, he argued, doesn't reason. It mirrors:



"It's just scraping from a lot of media noise and there's a recency bias on that. They say intelligence — it's not intelligent analysis. It's more like statistical collation."



He gave three concrete examples:

The World Cup Problem

Before the tournament, AI hedged on who would win. After Messi scored three goals in early matches, the AI pivoted entirely: Argentina will win. "It's just scraping from all the stuff people are posting on the web, collating that. But there's really no real nuanced analysis of how long this is going to endure, is one an aberration." The AI was a mirror of media sentiment, not an analytical engine.

The Watch Insurance Problem

He photographed a watch for insurance pricing. The AI quoted roughly 2x purchase price. Defensible as replacement cost, but never asked the critical question: "How much are you willing to spend for insurance? If you want to insure something for twice your purchase price, the premium is going to be twice what you should be paying." The AI solved the surface problem without understanding the tradeoff.

The SpaceX IPO Problem

Pre-IPO: wildly bullish. Post-IPO, as media narrative shifted to "100x revenue is ridiculous," the AI's tone shifted with it. "The tone is changing simply because there's so much media interaction and everything that has been posted that it's just picking up from that." The AI wasn't analyzing SpaceX. It was mirroring the media cycle.

For an institutional investor whose edge depends on thinking differently from the crowd, a system that converges on the crowd's opinion is useless, or actively dangerous.



The Three Failures

Beyond the epistemic problem, Udayan identified three specific failures that made generalist AI unusable for his actual workflow:

No memory

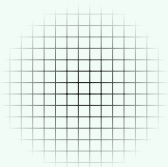
Bloomberg's "Ask B" function holds results for a limited time then clocks it off. He wanted a personal repository that remembers his research: the sequence of alterations, the reasoning chain, the in-sample vs. out-of-sample iterations.

No quality control

"It's just scratching the internet. Some of the stuff it puts out may or may not be the best quality. Sometimes the nuances are lost." When the AI is a statistical mirror of internet consensus, there's no mechanism to catch when the mirror is wrong.

No learning

The AI doesn't internalize your investment philosophy, your methodology, your way of thinking. Every interaction starts from zero. For someone running GARCH models and Black-Litterman optimizations, that makes it useless.



"If all I'm saying is just using variance optimization, everybody's saying that, and everybody's going to have the same portfolio at the end of the day. It's not a solution that develops an edge for you. It becomes quickly a zero alpha exercise."

The Search

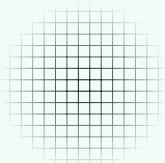
Udayan wasn't actively looking for a new AI platform. He had just returned from a week of conferences and was, in his words, "conferenced out." His colleague Ben Jacobson called to catch up, heard Udayan's skepticism about AI, and suggested a conversation about AdvisorClaw. Udayan agreed, mostly out of professional courtesy.

He approached it with skepticism. He'd seen enough AI demos to know that most of them overpromised. His guard was up.

But when Ben Jacobson opened the call not with a pitch but with a question: "What frustrates you about AI today?" Udayan started talking. And didn't stop.

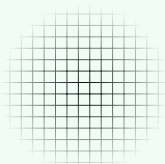
The GARCH Moment

When Ben pressed: "If it wasn't a generalist, if it had memory, if it could learn the way you like to do things, what else would you have it do?" Udayan went deep:



"There's a technique called GARCH — generalized autoregressive heteroscedasticity. You don't blindly use historical volatility over 10 years. You're modeling economically the rate of change. The best estimate of future vol is not implied vol in the options market — it's a measure of historical vol with the momentum drift. There's some core time correlation between what vol is today and vol tomorrow — autocorrelation."

Then he described exactly what he wanted:



"If it can do it well, then I can say: six months ago the optimal portfolio looked like this. Now that we had data come in, can you just use a GARCH model and update the parameters? Once you update the parameters, the correlation matrix changes, and it spits out: your portfolio looked like this in January, the optimization looks like this right now. And that becomes an automatic process."

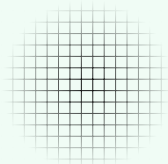
He tied it together with Black-Litterman, the optimization model that takes market weights and adds a "views matrix." If you do this sequentially, the system learns how your views evolve. It becomes proprietary to you. Generalist AI can't do that.

The Ask

Ben didn't sell. He validated. "This is the great financial crisis — take a bunch of bad loans, put them all together, and it won't be so bad if you average them all out. Being a generalist is nowhere near good enough. What you're describing is exactly what we built."

Then he anchored the solution in one line: "Two to three graduate-level employees that learn from you, get better each month, for the compute cost of lunch."

Udayan didn't need a next step proposed to him. He asked:



"Can I come to your thing one day and just have a look at it real quickly? Just a live demo? I don't want to see it on Zoom. To be honest, I just want to sit down with you guys."

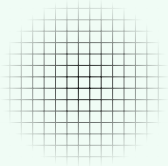
The screenshot displays the AdvisorClaw interface. On the left is a 'WORKSPACE' sidebar with folders like 'trash', 'Client', 'IDX info', 'outputs', 'reports', 'Research', and 'Shared'. The main panel shows a 'MARKDOWN DOCUMENT' titled 'Tax Loss Harvesting Analysis — Sarah Chen', prepared by 'IDX Insights Portfolio Analytics' on May 7, 2026. The report includes an 'Executive Summary' based on a review of Sarah Chen's taxable portfolio, noting a narrowed materially since the client's April 23 data upload. It lists three candidates: VNOQ (Vanguard Real Estate ETF) with a small gain, TSLA (Tesla) with a significant gain, and MSFT (Microsoft) with an unrealized loss of \$4,325. The summary concludes with a net harvestable loss of \$3,604 and estimated tax savings of \$1,489. The right sidebar shows 'AGENTS' with roles like 'IDX (main)', 'news', 'Market Researcher (market)', 'Portfolio Analyst (port)', 'webchat-g-agent', and 'Compliance Officer (compl)'. At the bottom right, a 'No scheduled tasks yet' message is displayed.

28 minutes. No deck. No feature list. A CIO who was "conferenced out" asking to drive in for an in-person demo.

He wanted to see it with his own eyes.

The Build

The demo took place on July 14, 2026. Ben McMillan joined Ben Jacobson for a 54-minute Teams session with Udayan. McMillan opened by framing the philosophy:



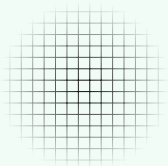
"AI is like a smart kid who graduated college but doesn't know Excel. AdvisorClaw sent him to business school, taught him Python, PowerPoint, and finance. Now we're turning him over to you to train on your company."

Then he got to work. What follows is the verbatim workflow McMillan walked Udayan through, the same workflow captured in AdvisorClaw's "Building a Volatility Model" training video.



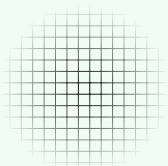
Step 1: Upload the Data (00:00–03:20)

McMillan had Udayan send over his Apex Equity Growth strategy returns. He hadn't seen the data before the call. He uploaded it to a fresh agent and typed:



"AI is like a smart kid who graduated college but doesn't know Excel. AdvisorClaw sent him to business school, taught him Python, PowerPoint, and finance. Now we're turning him over to you to train on your company."

The agent spent roughly a minute accessing the file, then responded:

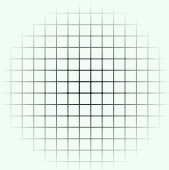


"Yep, here it is. It's monthly data. There's actually two return streams in here."

McMillan hadn't known it was monthly, or that there were two streams. The agent figured it out. Udayan was watching.

Step 2: Verify Understanding (03:20–03:48)

Before running the model, McMillan paused to verify Udayan's comfort with the methodology:



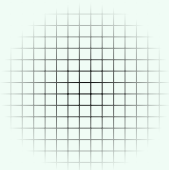
"Before we begin, please explain to me in simple terms what a GARCH model is."

The agent produced a clear explanation of GARCH(1,1), the autoregressive and moving-average components, why it's preferred over naive historical volatility, the Nobel Prize-winning lineage (Engle, 1980s, now at NYU). Udayan, an econometrician who applies advanced statistical models to financial data, saw the system pass a basic competence check.

McMillan's note in the video: "If you've heard about GARCH, maybe you want to try this in your own AdvisorClaw box. You can just chat with it about what the volatility model is, why it's important, why people like it. It's very popular in the quant space. We've used it for years."

Step 3: In-Sample Calibration (03:48–06:06)

The real work began:

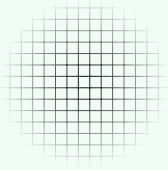


"Okay, great. Now I'd like to run two models: one full in-sample model on each return stream, and then a walk-forward model."

The agent calibrated GARCH(1,1) in-sample on both return streams. It used Python with pandas. "Because we tell it that's what you want to use anytime you're using quant finance," McMillan noted, and produced:

- GARCH omega, alpha, and beta parameters for each stream
- Half-life of volatility shocks
- Long-run (unconditional) volatility estimates

Then it interpreted the results unprompted:

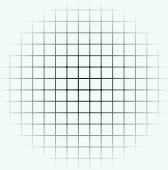


“The skewness is low, which suggests this is a managed series.”

Udayan's strategy is an actively managed stock portfolio. The agent was identifying characteristics of managed returns: low skew, serial dependence, from the data alone. McMillan, who's used GARCH for 20 years, confirmed the agent was producing PhD-grad-student-level output.

Step 4: Walk-Forward Validation (06:06–07:07)

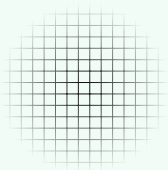
McMillan didn't stop at in-sample. He wanted out-of-sample validation, the standard a trained statistician like Udayan would demand:



“Yep, that sounds great. Let's proceed.”

The agent proposed a walk-forward plan: calibrate on the first 100 observations (~10 years of monthly data), validate on the remaining 50, then expand the window. This is how you test whether the model actually works out of sample, whether it would have predicted volatility in real time.

The agent went further than asked: it ran five model variants, not just the two McMillan requested. Standard GARCH(1,1). EGARCH (exponential, captures leverage effects). GJR-GARCH (treats upside vs. downside volatility differently). When McMillan asked what GJR was (he'd forgotten the acronym). The agent explained it immediately.



“I had actually forgotten what GJR was, so I asked it. It's creating all these artifacts, and it's telling me GJR is focused on treating upside volatility versus downside volatility differently. So it's already honing in on what the standard is for GARCH.”

— Ben McMillan, in the training video

Step 5: Output Generation (07:07–08:55)

The agent didn't just print numbers. It produced a complete analytical package:

- **Charts:** Conditional volatility plots for each model variant, walk-forward diagnostics comparing predicted vs. realized vol
- **Summary markdown file:** A formatted executive summary with key takeaways
- **Walk-forward diagnostics:** Out-of-sample error metrics, parameter stability plots
- **Autocorrelation analysis:** 1-month, 3-month, and 6-month lag autocorrelations on monthly returns

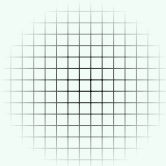
All from an upload McMillan hadn't seen before the call. All in under 10 minutes.

Live Critique: The Critic Subagent

McMillan then spawned a second agent, a GLM 5.2 subagent, and gave it one instruction: review the GARCH agent's work and flag anything questionable.

The critic caught that these appeared to be actively managed equity returns and flagged a methodological concern. Active management can introduce artificial serial dependence. Managers targeting stable volatility, using consistent rebalancing rhythms, or applying risk controls can suppress natural return clustering and distort autocorrelation patterns. This matters for GARCH, which assumes volatility dynamics are market-driven, not manager-driven. The GARCH agent hadn't raised this. The critic did.

Udayan watched an AI system catch a methodological issue that would matter in practice. This wasn't "statistical collation." It was second-order reasoning.

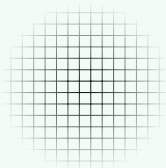


"I like to see the reasoning. I really do."

— Udayan, during the demo

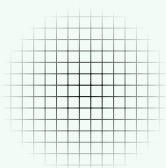
From Folder to Dashboard

McMillan pointed out the agent's file system — empty at the start of the call, now populated with charts, markdown summaries, and Python scripts:

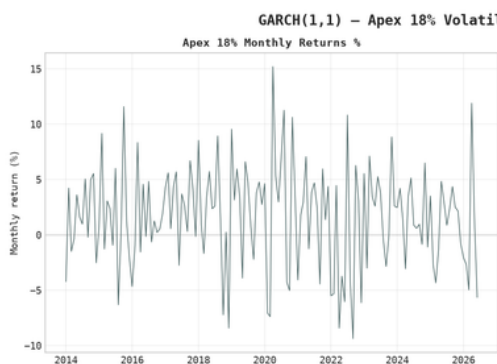


"This folder was blank when we started. All I did was upload these Excel returns, and now it's going through and creating all these charts. And so you can go through and have it do whatever you need it to."

The GARCH agent's outputs are reusable code. Run it once, and you can run it on a cron job every month, or turn it into an interactive dashboard. McMillan noted that clients have done both.



"From here you can turn this into something in implementation that runs on a cron job every month if you wanted to. We've had people turn these into interactive dashboards — and all this was 10 minutes of work."



GARCH(1,1) – Apex 18% Live Fit
Sample: 2014-01-01 – 2026-06-01
Trading months: 150

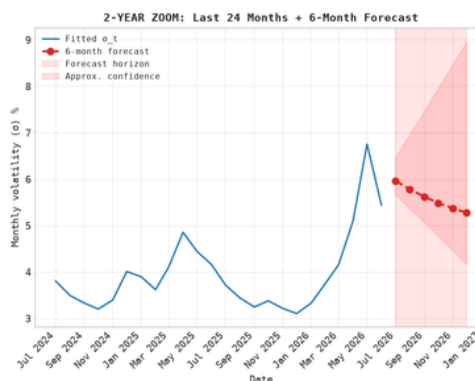
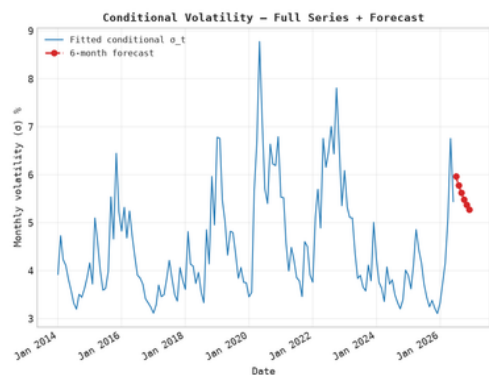
Param	Value
mu	1.0495
omega	3.782245
alpha	0.2690
beta	0.5640
awb	0.8329
LT vol	16.40% ann.

6-Month Vol Forecast (ann.)

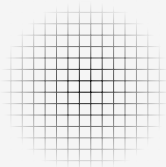
Month 1:	20.66%
Month 2:	20.03%
Month 3:	19.48%
Month 4:	19.01%
Month 5:	18.61%
Month 6:	18.27%

Converged: True
Log-likelihood: -433.72

Udayan reviews his models every six months. A cron job that recalibrates GARCH parameters, updates the correlation matrix, and feeds into Black-Litterman optimization is exactly the automated workflow he described wanting in the first call.



Udayan's Reaction



"This is fantastic. This is exactly the quick way of doing things that save me a lot of time rather than going through SPSS, doing my own GARCH."

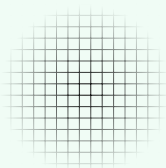
"This is actually good. I mean, that's what I figured it would do — the Markov chain type of analysis."

"Send this chart to me by email if you don't mind, just as a one pager."



Then, the moment that changed the conversation:

Udayan realized the demo was running on his own provisioned box.



"This is on my computer?"

— "Yes, *your provisioned box.*"

By the end of the call, Udayan was logged into `scottsdale.advisorclaw.ai` himself, changing the appearance to light mode, increasing the font size, and exploring the file system and agents on his own.

The Tool

What Udayan saw was not a vendor product demo. It was a live build on his own data. Here's what emerged:

GARCH Volatility Engine

The agent calibrated five GARCH variants on Udayan's Apex Equity Growth returns:

Autocorrelation & Regime Detection

Component	What It Produced
GARCH(1,1) standard	Omega, alpha, beta parameters; half-life of vol shocks; long-run unconditional vol
EGARCH	Leverage-effect modeling: asymmetry in how positive vs. negative returns affect future vol
GJR-GARCH	Upside vs. downside volatility separation, the industry standard for risk modeling
Walk-forward validation	Calibrated on 100 obs, validated on 50, expanding-window methodology
Model diagnostics	Out-of-sample error metrics, parameter stability across windows

- **Serial autocorrelation:** 1-month, 3-month, and 6-month lag analysis on monthly returns, testing Udayan's hypothesis that good management teams produce discernible serial dependence
- **Markov-switching regime detection:** Identified distinct volatility regimes in the return series, with transition probabilities



Critic Review Layer

A second agent (GLM 5.2) reviewed the first agent's output and flagged:

- Apparent actively managed equity return characteristics (low skew, serial dependence)
- Methodological concern — active management can introduce artificial serial dependence that distorts GARCH assumptions
- Recommendations for adjusted GARCH specifications to account for smoothing bias

Deliverables (Generated in Under 10 Minutes)

- Conditional volatility charts for each model variant
- Walk-forward diagnostic plots (predicted vs. realized vol)
- Formatted markdown executive summary with key takeaway
- Reusable Python scripts, runnable manually, on cron, or as an interactive dashboard

The Architecture

- **Open chassis.** Udayan's own AdvisorClaw box, provisioned at scottsdale.advisorclaw.ai. No shared tenancy. His data, his models, his agents.
- **Open-source models.** DeepSeek v4 Pro (default, verbose, strong reasoning), Kimi K2.6 (programming backup), GLM 5.2 (critic agent). No per-token vendor costs eating into margin. McMillan noted he used to spend \$15K/month on frontier models; open-source cut that to tens of dollars.
- **mem0 long-term memory.** SOC 2 compliant, dedicated instance. Tracks in-sample vs. out-of-sample iterations, parameter changes, and reasoning chains over time, the institutional memory Udayan demanded.
- **Agentic harness.** "The model itself is only 30–50% of the utility. At least 50% of the value is the agentic harness: the tools the model has access to, how it knows to use those tools. Especially in finance, this is why we spent years building this out." — Ben McMillan

The Results

Quantitative

Metric	Before	After
Time to GARCH calibration (in-sample + walk-forward, 5 variants)	Hours in SPSS	Under 10 minutes
First call to demo request	N/A, wasn't looking	28 minutes
Demo to "send me the chart"	N/A	54 minutes
Model variants tested	1 (manual, SPSS)	5 (automatic, including EGARCH and GJR)
Quality review	Self-review only	Independent critic subagent caught desmoothing concern
Institutional memory	None. Bloomberg "Ask B" clocks off	mem0 persistent memory tracking all iterations
Cost per GARCH run	Frontier models: variable	Open-source IDX models: ~\$0.14

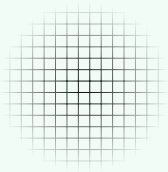
Qualitative

From skeptic to champion in two meetings. Udayan went from "it's statistical collation" to "send me the chart" to "let's get this in front of my advisors" without a single slide deck, without a pricing conversation, without a six-month evaluation cycle.

Live editing changes the dynamic. When Udayan asked a question, McMillan typed it into the agent and the output updated. The meeting became a collaborative analytical session, not a vendor presentation.

The critic subagent closed the trust gap. Udayan didn't have to take McMillan's word that the GARCH output was sound. He watched a second AI agent independently review the first agent's work and catch a real methodological issue. That's not marketing. That's evidence.

He logged into his own box. By the end of the demo, Udayan was using the system himself, changing settings and exploring the file structure. He wasn't evaluating a product. He was using it.



"This is wonderful. I'll play around with all this."

— Udayan, at the end of the demo

The Pivot: From Power User to Fleet Deployment

At lunch after the demo, Udayan shifted the conversation. His personal GARCH use case is real, and he was already thinking about how he'd use it. But the larger opportunity is getting AdvisorClaw in front of the RIA's 34 advisors.

Udayan was clear about what would work for his team. These advisors are relationship-driven professionals. They build trust with clients over years, not quarters. Their value is in understanding family dynamics, tax situations, estate complexity, not in running volatility models. They need tools that amplify what they already do well, not tools that require them to become something else.

They operate in portfolio frameworks clients understand. They translate market noise into plain language because that's what their clients need. Any new tool has to fit that workflow — not because the advisors can't handle complexity, but because their clients shouldn't have to.

The firm also positions itself as a multi-family office, and Udayan is always looking for tools that support that holistic approach. Many of his advisors came from wirehouses or smaller RIAs where they never had access to true family-office infrastructure. The right tools can help them deliver that experience without forcing them to become technologists.

Ben's response: don't lead with quantitative firepower. Build something that any advisor can pick up immediately — simple on the surface, powerful underneath. Something that feels like an upgrade to their existing process, not a replacement of their expertise. Go through the advisor trainer, who knows what the team will actually use.

Udayan agreed. He's bringing the company's advisor trainer to the next lunch to see what might work. The GARCH power user became the fleet-deployment champion.

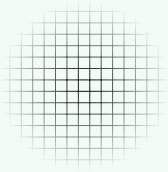
The Bigger Picture

This story isn't really about GARCH models.

Every sophisticated investor has a methodology. Udayan's is GARCH volatility modeling with walk-forward validation, Black-Litterman views matrices, and Markov-switching regime detection. Another CIO might lead with factor models, stress testing, or co-integration analysis. The point isn't which methodology. The point is that AdvisorClaw can build to any methodology.

Udayan spent months frustrated with generalist AI that couldn't remember, couldn't learn, and couldn't reason beyond internet consensus. He described what he wanted in his own words, in a 28-minute first call. He watched it get built live on his own data in under an hour. He logged into his own box during the demo.

The sequence of value delivery was compressed to a single meeting. No RFP. No procurement. No six-month pilot. A CIO articulated a sophisticated quantitative workflow, watched it materialize on his own strategy returns, and asked for the output by email, all within 54 minutes.



"What I would like to have is that memory kind of thing embedded. So the sequence of alterations and thinking that we went through can be revisited at any one time. I'm a traditional statistician so it's an out-sample in-sample type of exercise. If I hadn't done this, would I have had a better fit?"

That's not just a feature request. It's the thesis of AdvisorClaw in the client's own language. A system that remembers your methodology. Tracks your iterations. Validates your models out of sample. Catches its own errors. Gets smarter within your framework.

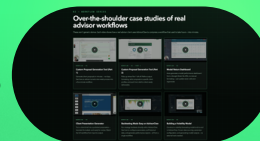
If a quantitatively skeptical CIO can see his ideal workflow built live on his own data in under an hour, what can you build on yours?



Visit Udayan Mitra's Linked in page:
<https://www.linkedin.com/in/udayanmitra/>



For video walkthroughs of this tool, visit:
advisorclaw.ai/resources



The views expressed by Udayan Mitra are his own and do not necessarily reflect those of his firm, its management or its affiliates.

AdvisorClaw by IDX — advisorclaw.ai