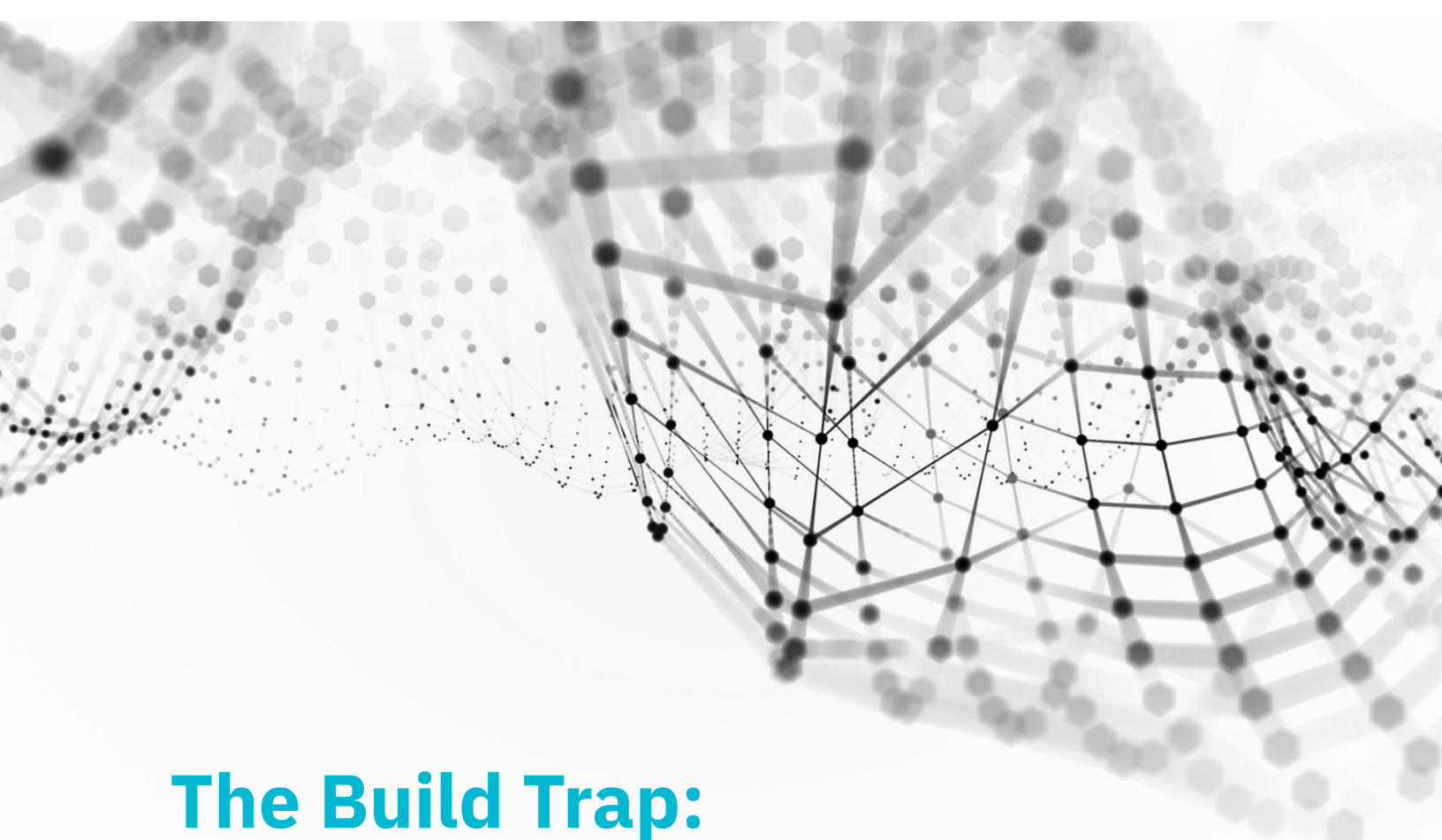




# **The Build Trap:**

## What We Learned from Surveying 45 Investment Data Leaders

Promises, Pitfalls, and Hidden Costs Based on a Comprehensive  
Survey of In-House Data Platform Initiatives



# Contents

## Executive Summary

### 1 The Unvarnished Reality: Key Performance Metrics

### 2 The Implementation Nightmare

### 3 The Seven Deadly Sins of In-House Builds

### 4 The Outcome: Projects That Did Not Deliver

### 5 The Financial Reality: Total Cost of Ownership

### 6 The AI Mirage: Why Coding Assistants Haven't Changed the Build-vs-Buy Equation

### 7 The Overwhelming Verdict: Buy, Don't Build

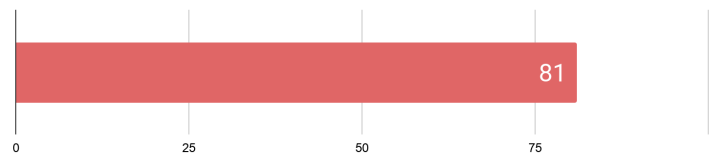
### 8 A Better Path Forward: What Vendor Platforms Deliver

### 9 Conclusion

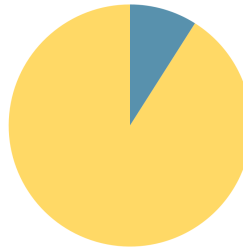
# Executive Summary

For investment firms, the promise of a custom-built data management platform is seductive: complete control, perfect fit, and perceived lower long-term costs. Our survey of 45 senior data, technology, and operations leaders - who collectively attempted to build their own platforms - reveals a radically different reality.

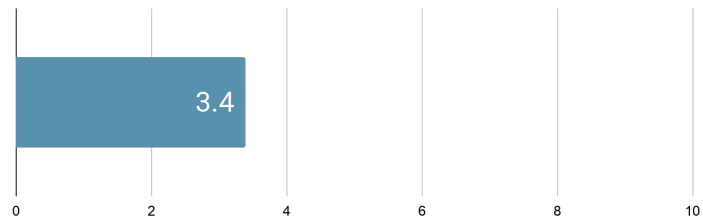
**81%** rated their implementation as “Very Challenging” or “Extremely Challenging.”



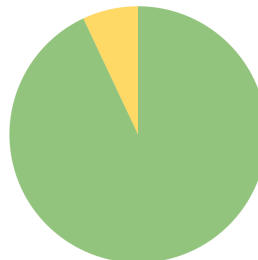
**Only 9%** reported that their project met original business objectives.



The average **Overall Platform Success** rating was just **3.4** out of 10.



**93%** of respondents (42 of 45) now explicitly say they would **buy a commercial solution** rather than build again.



The survey also revealed a consistent pattern: firms underestimated the complexity of building and maintaining an enterprise data platform, while failing to fully appreciate the operational, governance, and cost advantages offered by established vendor solutions.

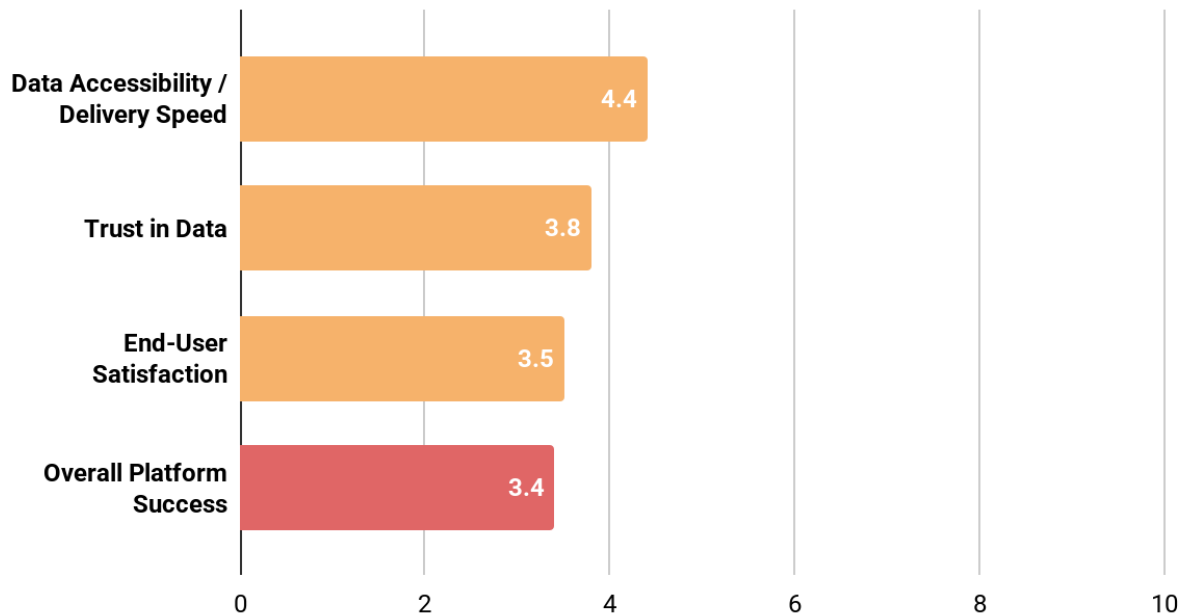
| What firms that build underestimated   | What vendor platforms provide                 |
|----------------------------------------|-----------------------------------------------|
| Integration complexity                 | Proven connectivity and data models           |
| Operational usability                  | Purpose-built user experiences                |
| Governance and compliance requirements | Built-in controls, lineage, and auditability  |
| Delivery risk and scope creep          | Mature functionality and faster time-to-value |
| Long-term cost and maintenance burden  | Predictable ownership and ongoing support     |

The in-house path consistently led to significant delays, budget overruns, brittle architectures, and - most critically - a pervasive lack of trust in the very data the platforms were designed to deliver. This whitepaper distils their experiences into a clear, evidence-based case for a buy-first strategy in investment data management.

## 1 The Unvarnished Reality: Key Performance Metrics

We asked respondents to rate four dimensions of their platform's performance on a scale of 1 to 10. The results paint a stark picture of underperformance.

### Performance Metrics



#### Trust in data - the foundation of any investment process - scored below 4.

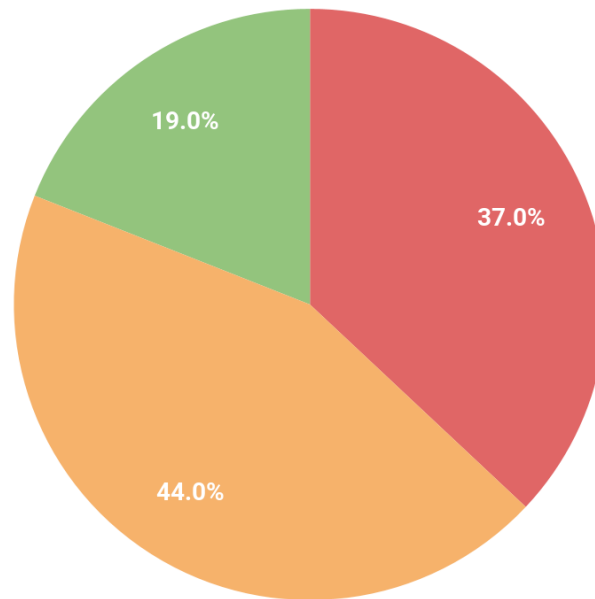
Even when the technical infrastructure worked, the output frequently failed to meet the standards of portfolio managers, risk teams, and compliance officers. As one Head of Quantitative Research noted,

*“We now have a **beautifully fast pipe** delivering garbage data to the front office.”*

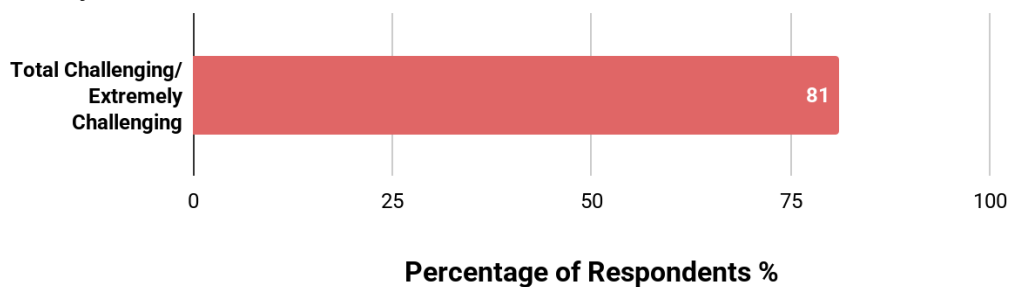
## 2 The Implementation Nightmare

The difficulty of building a modern, enterprise-grade data platform was universally underestimated. When asked to characterize the implementation, the distribution was overwhelming.

- Extremely Challenging
- Very Challenging
- Moderately Challenging



### Difficulty Level

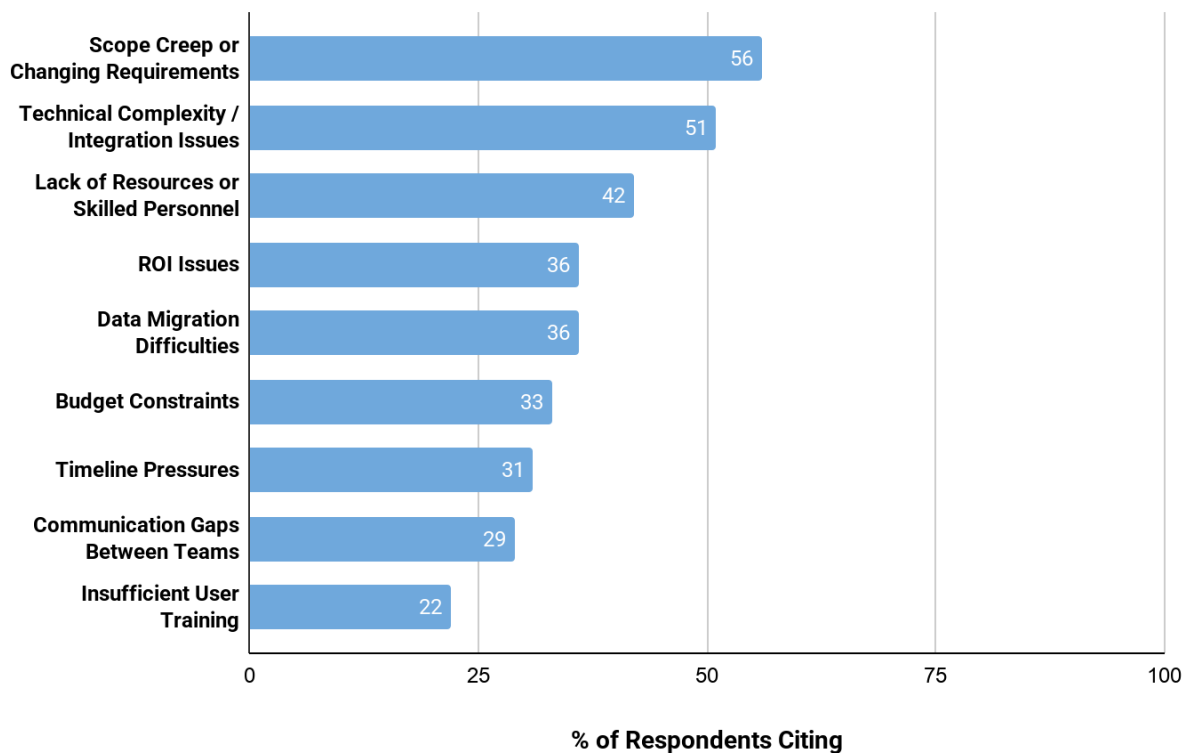


This difficulty translated directly into delivery failures. Only three respondents (7%) reported an on-time or near-on-time delivery. **Over half (53%) experienced significant delays**, often measured in years rather than months.

### 3 The Seven Deadly Sins of In-House Builds

The multi-select survey question on key challenges reveals the most common pitfalls. The following table ranks the obstacles that plagued the vast majority of projects.

#### Challenge



**Scope creep was the single most common killer of in-house initiatives.** As the COO of a mid-sized asset manager explained,

*“The business kept adding requirements after sign-off with no change control process. We ran out of budget 14 months in and had to descope the multi-eye approval workflows and exception management entirely - which were the two features operations most needed.”*

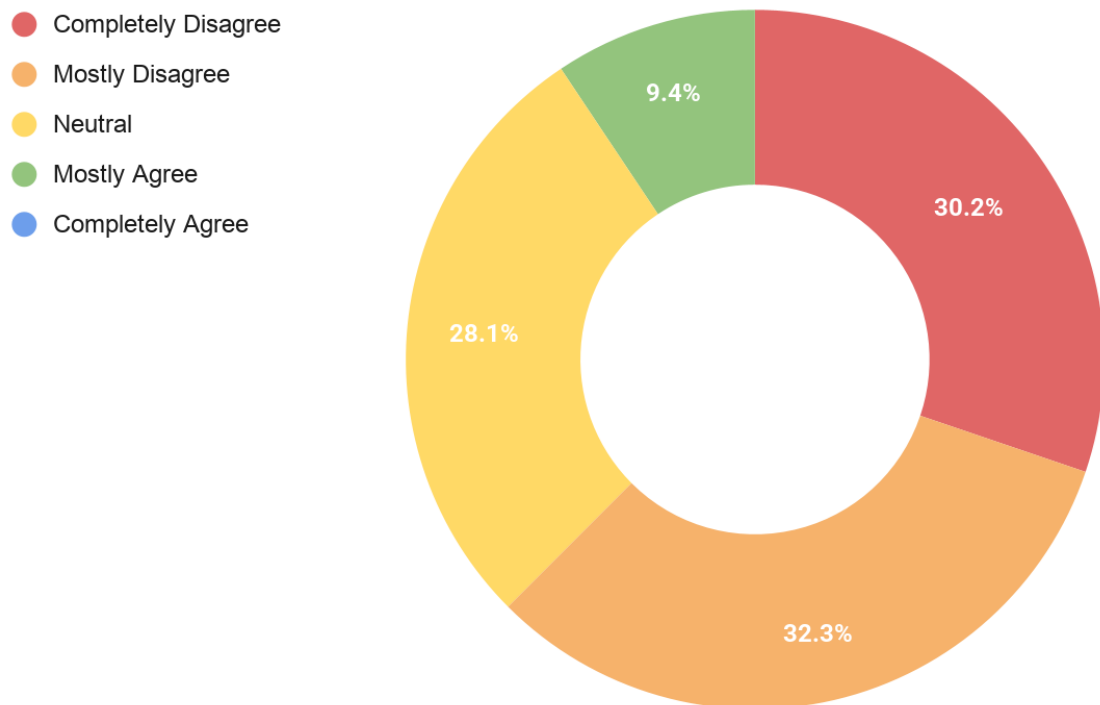
**Technical complexity was a close second.** Firms repeatedly underestimated the difficulty of multi-vendor data matching, corporate actions processing, and building intuitive exception-management UIs. A Lead Data Engineer summarized it bluntly:

*“We’re back-end engineers; we don’t know how to build a high-performance, intuitive UI for data stewards. What we delivered is a clunky SQL-driven nightmare that the business refuses to use. They still manage breaks in Excel.”*

## 4 The Outcome: Projects That Did Not Deliver

### Meeting Business Objectives

When asked whether the project met its original business objectives, the responses were damning.

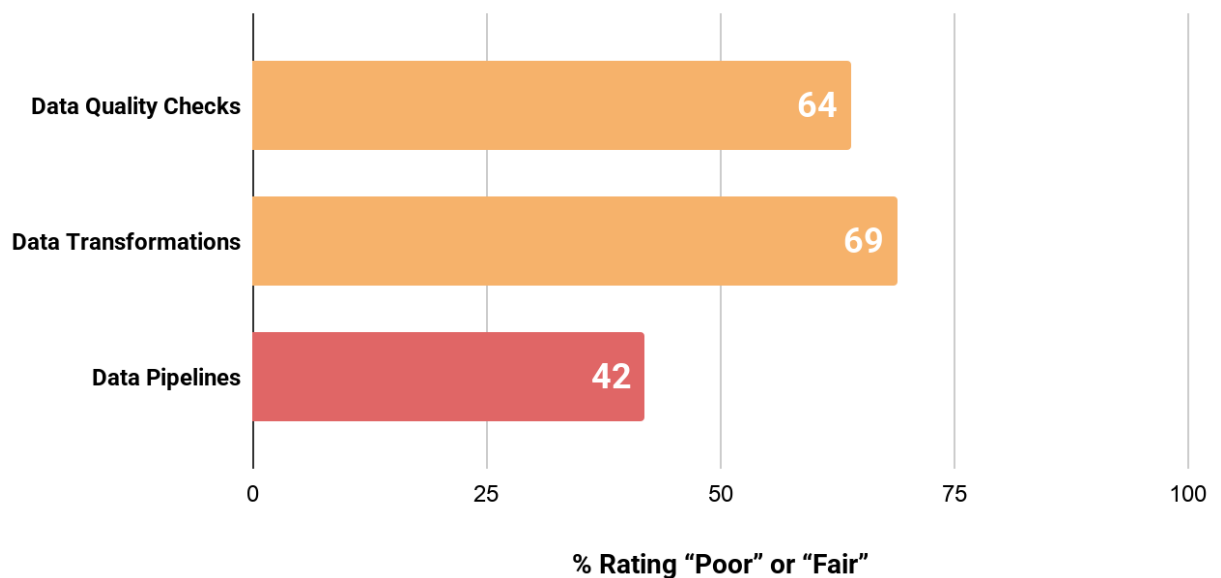


Only four out of 45 firms (9%) could mostly agree that they achieved what they set out to do. For the other 91%, the platform either missed the mark entirely or left the organization in a state of costly ambivalence.

### Documentation Quality

The build process also fell short on documentation - a critical asset for audit, maintenance, and business continuity. Respondents were asked to rate the quality of documentation for data pipelines, data quality checks, and data transformations.

#### Documentation Area



Poor documentation was not just a housekeeping issue; it created key-person risk and made regulatory reporting a recurring crisis. A Chief Compliance Officer described the consequence:

*“We have no meaningful end-to-end data lineage, making regulatory reporting a manual, nerve-wracking nightmare.”*

## 5 The Financial Reality: Total Cost of Ownership

The initial business case for building typically emphasized lower year-one licensing costs. The survey reveals that this calculation systematically ignored the true total cost of ownership (TCO).

- **Budget overruns** were common, with one Director of Data Management reporting a final cost of **2.3x the original budget** after accounting for staff turnover and rework.
- **Ongoing maintenance** consumed significant engineering capacity. A Head of Operations Technology noted, *“Ongoing maintenance is consuming 1.5 FTE of engineering time that wasn’t budgeted.”*
- **Key-person risk** created a hidden liability. Multiple respondents described their platform as dependent on one or two developers whose departure would render the system unmaintainable.

As one Director of IT concluded,

*“On a five-year view, a vendor platform would likely have been cost-neutral and delivered more functionality. The build vs. buy decision at our firm was made on first-year costs only.”*

## 6 The AI Mirage: Why Coding Assistants Haven't Changed the Build-vs-Buy Equation

A natural question in any boardroom debate is whether the advent of generative AI coding tools fundamentally alters the economics of in-house development. If AI can write boilerplate code, generate SQL queries, and accelerate feature delivery, doesn't the build option become more viable?

The collective experience of our 45 respondents suggests the answer is a decisive **no**.

As one Head of Data Architecture, who experimented extensively with AI copilots during his firm's build effort, reflected:

*"The more complex a domain and system, the more AI falls completely short. Our management assumed AI would let junior developers produce enterprise-grade code, but it was the exact opposite. The AI confidently generated plausible-looking changes to matching logic, only for the performance over large volumes of data to be completely degraded. Only a senior engineer with deep buy-side domain knowledge could spot many of the errors. The tools accelerated the production of code, but without expert oversight they accelerated the production of bugs."*

This observation cuts to the heart of why AI has not made the in-house build path safer for asset managers, and is unlikely to do so in the foreseeable future.

- **Domain complexity is non-linear.** A system that works for three vendor feeds will not scale smoothly to fifteen. The edge cases in security mastering, corporate actions, and cross-reference management increase complexity exponentially. AI assistants, trained

on general-purpose code, have no model of these esoteric financial data structures and can easily hallucinate plausible but dangerously incorrect logic.

- **The bottleneck is understanding, not typing.** The survey's most-cited challenges - scope creep, multi-vendor data matching, exception management design - are problems of *requirements definition* and *domain expertise*, not of coding throughput. AI can write a function; it cannot negotiate a shared definition of "data quality" between a portfolio manager and a risk analyst.
- **AI is a tool for experts, not a substitute for them.** The firms that succeeded in building something useful were those with experienced data architects who used AI carefully and sceptically. For the majority who struggled with resource constraints and junior teams, AI would likely have compounded the problem by increasing the volume of unaudited, brittle code.

The typical management response - *"It's only a matter of time until AI gets much better at complex systems"* - may prove true, but the timeline is uncertain and the cost of waiting is real. As the same architect put it:

*"Complexity is non-linear. AI might get there eventually, but I'm not betting my firm's data infrastructure on a five-year hope. By the time the tools are genuinely reliable for this domain, the vendor platforms we could have bought today will have ten more years of embedded expertise we can't replicate."*

The verdict from the survey is consistent: AI assistants do not alter the fundamental build-vs-buy calculus. They make it faster to write code - but the hard work of designing, governing, and maintaining a trusted data platform remains a deeply human, deeply specialized challenge. A vendor platform arrives with decades of that domain reasoning already baked in. An AI-augmented in-house team, by contrast, must still learn it from scratch, one painful edge case at a time.



## 7 The Overwhelming Verdict: Buy, Don't Build

The open-text question “What Would You Change?” produced a near-unanimous response. The following word cloud representation captures the essence of the 45 answers.

---

**Buy · Vendor · Off-the-Shelf · Commercial Platform · Don't Build ·  
Configurable · SaaS · Managed Service**

---

A selection of direct quotes reinforces the strength of this conviction.

- *“Buy. Don't build. The buy-side data management space has mature vendors with deep domain expertise.”* – **COO, ABC Fund Services**
- *“I would never again underestimate the value of vendor-driven automated lineage and impact analysis. Manual documentation is a lie.”* – **Chief Data Officer, Large Pension Fund**
- *“We would engage a managed service or buy a platform with a curated feed library. Our business is lending money, not mapping CSV files.”* – **CTO, Private Credit Direct Lender**
- *“Asset managers shouldn't pretend to be enterprise software development firms.”* – **COO, Hedge Fund**

The message is unambiguous: the project these leaders would change is the decision to build in the first place.

## 8 A Better Path Forward: What Vendor Platforms Deliver

The survey responses implicitly define the seven capabilities that a mature, off-the-shelf data management platform must provide. These are the areas where in-house builds consistently failed, and where a strategic buy decision delivers immediate, compounding value.

| What In-House Builds Got Wrong                                                                                   | What a Vendor Platform Provides                                                                                                                                                            |
|------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Multi-vendor data matching</b> consumed years of engineering time and still produced brittle results.         | Curated, maintained connector libraries and a canonical data model that handles cross-referencing out of the box.                                                                          |
| <b>Exception management UIs</b> built by backend engineers were universally rejected by operations teams.        | Purpose-built, configurable stewardship interfaces designed for buy-side workflows.                                                                                                        |
| <b>Data lineage</b> was an afterthought, bolted on late and delivering unusable “spaghetti graphs.”              | Automated, cell-level lineage and impact analysis that satisfies audit and regulatory demands.                                                                                             |
| <b>Governance and security</b> were either retrofitted or implemented so rigidly that users bypassed the system. | Configurable, role-based access controls that balance security with usability and are maintained as regulations evolve.                                                                    |
| <b>Scope creep</b> derailed projects without formal change control, leading to budget exhaustion.                | A feature-rich, configurable platform that eliminates the need to build commodity functionality, allowing firms to focus proprietary effort only on true alpha-generating differentiation. |

| What In-House Builds Got Wrong                                                                    | What a Vendor Platform Provides                                                                                                      |
|---------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>Staff turnover</b> created existential key-person risk for critical infrastructure.            | An established vendor with a broad ecosystem, support team, and documented product roadmap ensures business continuity.              |
| <b>Total cost of ownership</b> was systematically underestimated, often doubling initial budgets. | Predictable subscription pricing that includes ongoing maintenance, updates, and support - making true TCO transparent from day one. |

When a Head of Portfolio Analytics says, *“I’d buy, not build. The opportunity cost and the technical debt we’ve accrued by maintaining this monster have erased any perceived cost savings,”* he is describing precisely the trap that a vendor-first strategy avoids.

## 9 Conclusion

The evidence from 45 investment firms is conclusive: the in-house build of a comprehensive data management platform is a high-risk, low-reward strategy. The overwhelming majority of projects fail to meet their objectives, deliver late, blow their budgets, and produce systems that end users do not trust and will not adopt.

In almost every case, the respondents’ single most valuable outcome was the hard-won realisation that their firm should have bought a commercial solution from the outset. As one Chief Data Officer put it,

*“We now have a clear understanding of what we actually need from a data platform - which we’ll now buy rather than build.”*

**Don't let your firm become the next cautionary tale.** The investment data management market now offers mature, configurable, and deeply domain-aware platforms that have already solved the challenges these 45 teams spent years and millions trying - and failing - to overcome.

---

*For a confidential consultation on how a modern data management platform can accelerate your firm's data strategy without the risk of an in-house build, contact us today.*

## About Fencore

Fencore is a cloud-native, no-code, AI-driven enterprise data management software provider focused on serving the investment management industry. We make data management easy, affordable, and accessible to financial institutions of all sizes; and empower clients' employees via our intuitive, no-code platforms that help drive the success of their businesses.

Fencore is headquartered in Singapore, and serves markets globally.

Visit us at [www.fencore.co](http://www.fencore.co) to find out more about our solution and services, and request for a demo.