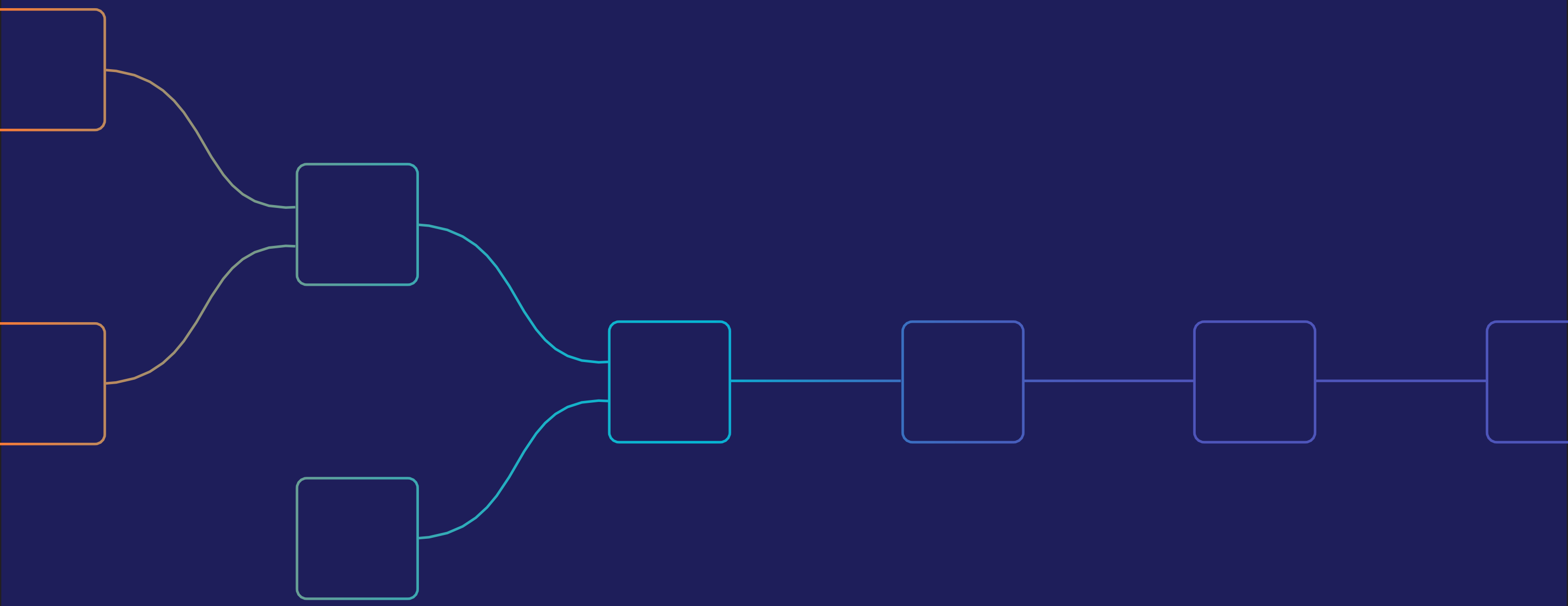
White Paper

The Rise of Agentic Data Preparation & Analysis

How AI agents empower business teams to develop and deploy workflows quickly, and why waiting is no longer an option.



Executive Summary

The way enterprise data teams work is about to change as fundamentally as software development did when AI coding assistants arrived. It took roughly 6 to 12 months for AI-first development to become the default for software engineers; data preparation is following the same curve.

The core shift: AI agents now generate data workflows from plain-language instructions. Business users are experts in the domain and understand their data well, but aren't deep in the platform stack. But this bottleneck that has defined enterprise data teams for a decade is disappearing. Modern platforms use AI to generate visual data workflows directly to production without waiting on data engineers to build and tune pipelines.

For data and technology leaders, three questions matter:

1. How can you capitalize on faster data preparation to drive better outcomes for your critical business processes and deliver more services?
2. Are your current tools capable of speeding up time to insights while being auditable, cloud-native, and AI-first?
3. What does a low-cost and low-risk adoption of AI for data preparation look like?

This white paper answers all three. The conclusion is clear: organizations that move to AI-native data preparation will build a structural productivity advantage and enable analysts to focus their extra capacity towards other high-value work. Those who wait will soon find themselves playing catch-up against their peers.

> *The playing field is poised to become a lot more competitive, and businesses that don't deploy AI and data to help them innovate in everything they do will be at a disadvantage.*
>
> **— Paul Daugherty, Chief Technology and Innovation Officer**

The Limitations of Desktop Tools for Solving Data Preparation Challenges

Business teams have data questions. Getting those questions answered requires data prep work: cleaning, joining, transforming, and validating. Data prep often requires scarce technical expertise. The resulting queue is a permanent feature of enterprise life, with analysts backlogged, business users waiting, and decisions made on stale or incomplete data.

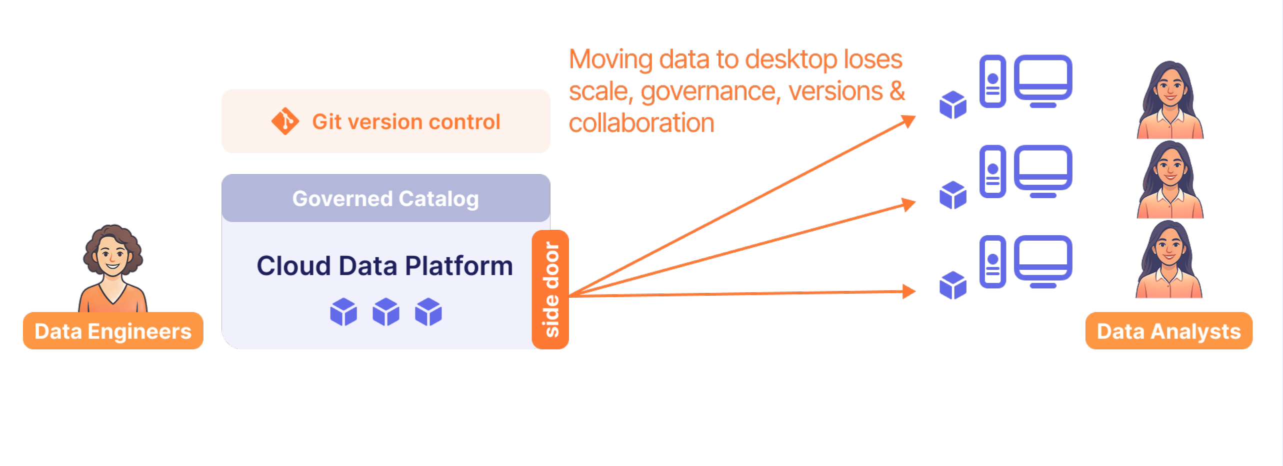
Desktop data preparation tools like Alteryx addressed part of this problem well. They gave analysts a visual, drag-and-drop environment to build data workflows without writing code. Users could move fast on their local machines, and for the era in which they were built, this was genuinely transformative.

But that era ended. Three shifts have exposed the limits of desktop architecture in ways that can't be patched around: the rise of cloud data platforms, the adoption of Git-based collaboration, and the arrival of generative AI.

1. Cloud data platforms changed where data lives and runs:

Databricks, Snowflake, and BigQuery now handle massive amounts of enterprise data. When analysts pull data to their desktops to work on it, they lose scale, governance, and version control, and they create security exposure. The data leaves the governed environment through a side door, and the business logic and processes become siloed.

The desktop data prep problem



2. Git changed how production software is built and deployed:

Every modern engineering team works with version control, branching, code review, and CI/CD. Desktop data workflows sit outside this model entirely. Promoting analyst work to production means handing it to a data engineer to rewrite, a hand-off that adds weeks and introduces drift between what was specified and what actually runs.

3. AI changed what's possible for everyone:

LLMs can generate high-quality data transformation code from natural language. This isn't an incremental improvement to existing tools; it's a new development mechanism that only works well when the tooling is built to take full advantage of it.

Desktop data preparation vendors aren't standing still. But their architecture, designed for local execution before the cloud, before Git, and before AI, constrains how deeply they can integrate these capabilities. Adding an AI chat layer on top of a legacy product is not the same as an AI-native architecture–the productivity gains are not equivalent.

Why AI Is Different Than Past Technology Changes

Data preparation has seen many waves of tooling change: better connectors, cloud deployment, distributed compute, self-service BI, and so on. Each time, incumbent desktop products adapted and survived. This time is different, for a specific architectural reason.

Data workflows are a specialized form of code:

This matters because AI is exceptionally good at generating code, and it is getting better every quarter. The same LLM capabilities that now write production software can generate, validate, and refactor data transformation logic. **Data prep sits squarely in the domain where AI produces its highest-quality output.**

But AI output needs to be validated for trust:

This is the critical insight that separates genuine AI-native products from AI-washed ones. A workflow that is “mostly right” is not useful. Incorrect data transformations produce incorrect outputs, which can propagate through downstream reports, dashboards, and decisions. AI can generate a pipeline, but what data workers need is the ability to efficiently verify that the pipeline is correct.

Visual pipelines are significantly faster and more reliable for catching errors than reviewing generated SQL or Python. A non-expert can trace a visual workflow and identify a wrong join or a missing filter, and can spot-check data records. The same person cannot reliably audit 200 lines of PySpark.

The right development loop for the AI era is Generate → Refine → Deploy

Generate

An AI agent generates a first-draft workflow from a goal description or requirements document, in minutes rather than days

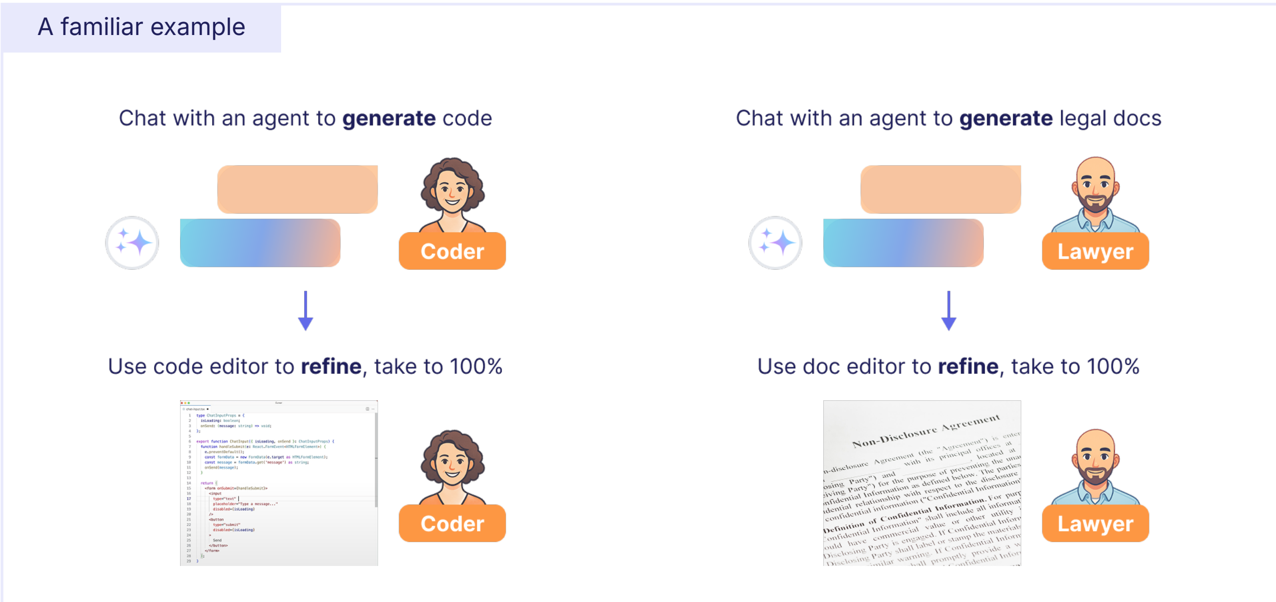
Refine

The user inspects the workflow visually, sees the data at each step, and catches and corrects any errors

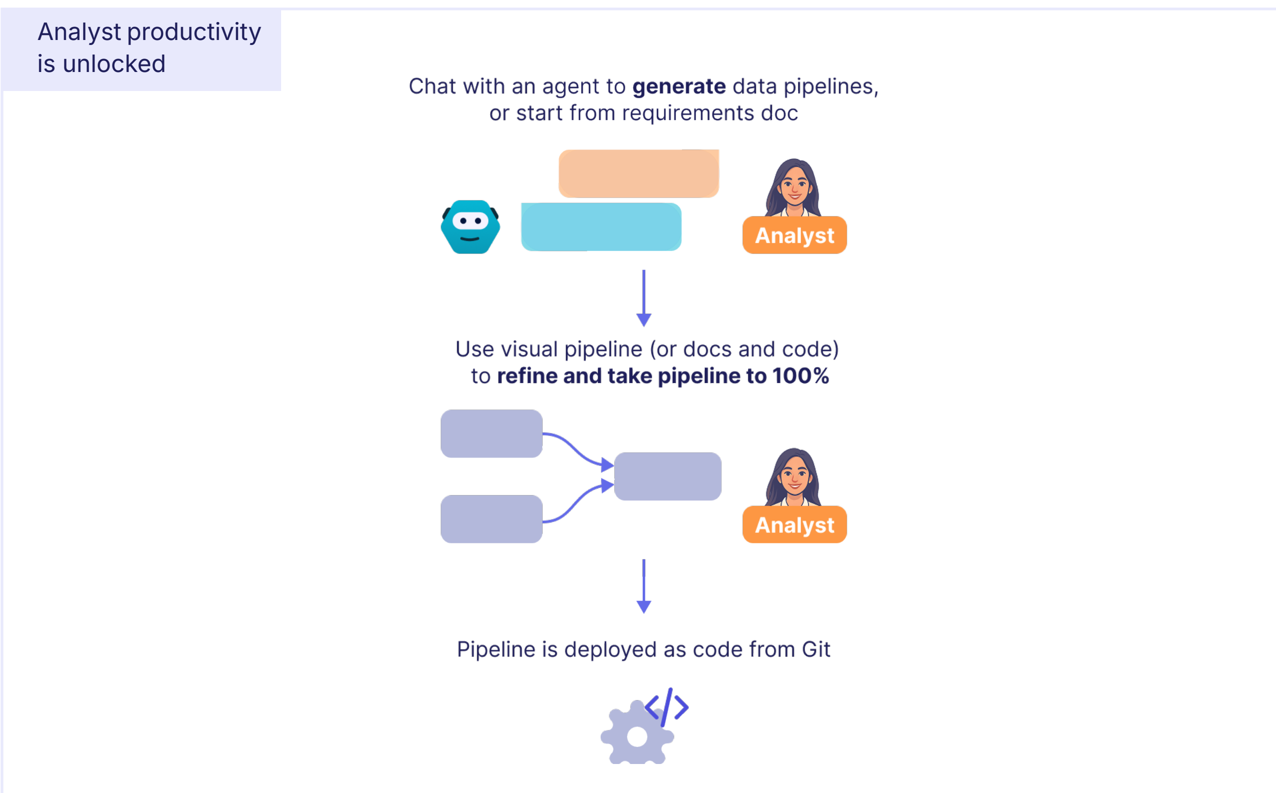
Deploy

The validated data workflow compiles to native SQL or Spark and deploys directly to the cloud data platform with no rewrite

This loop works for coders in software development, where AI generates code for developers to review in an IDE and merge to production, and for legal teams, where AI generates a draft for lawyers to review in a document editor and finalize.



For data teams, the equivalent is the visual pipeline editor. The draft is reviewable by the person who understands the business logic, not just the person who can read Python.



What Compound Agent Systems Deliver

Most AI claims in enterprise software today follow the same pattern: take an existing product, add an LLM-powered chat interface, and call it AI-native. The productivity gains from this approach are small. The workflow is still fundamentally the same, and the AI assists only at the margins.

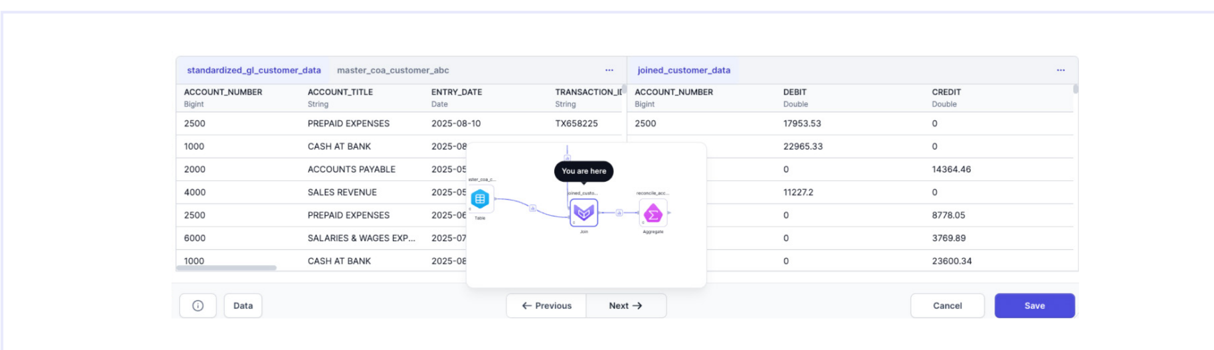
Genuine AI-native data preparation requires compound agent systems. These systems are coordinated networks of specialized agents, each tuned for a specific task and orchestrated by a manager layer that interprets intent and routes work appropriately.

A typical data analysis workflow involves a predictable sequence of narrow tasks: discovering the right datasets, joining and transforming them, validating outputs, documenting the logic, and generating tests. Each of these is a distinct problem that benefits from a different kind of AI specialization. A single general-purpose LLM call handles none of them particularly well. A purpose-built agent for each task, with the right data context loaded, the right skills in place, the right output format specified, and the right validation loop in place, handles all of them well.

The practical implications break down across five capability areas:

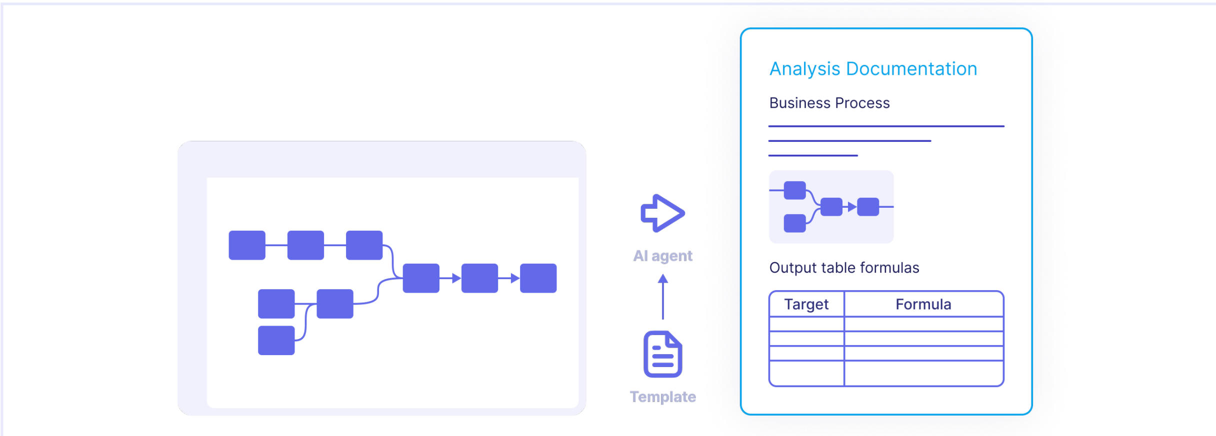
1. **Data discovery agents** search across all connected data platforms by attribute and meaning, not just table and column names. An analyst can ask "find me datasets related to customer payment history in Q3" and get relevant results across their cloud data platform with enough context to identify and explore the right one.

2. **Transformation agents** generate step-by-step workflow logic from natural language. Rather than generating an entire data workflow at once (which reduces accuracy as workflows grow), the reliable pattern is iterative: describe the next step, the agent generates one or two visual components, the user validates and refines if needed, and the process repeats. This maintains high accuracy through iteration.

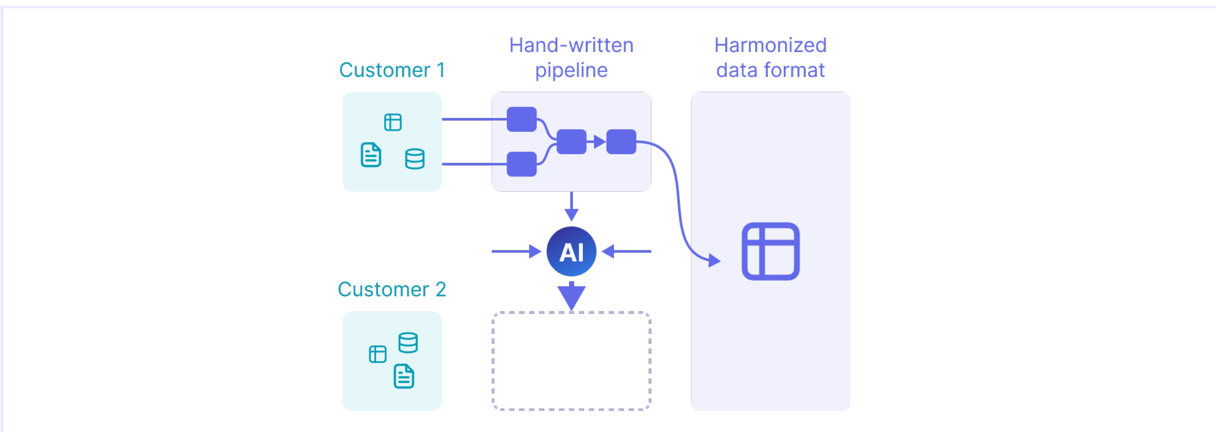


3. **Error-correction agents** detect issues in pipelines automatically and propose fixes. This is particularly valuable in production environments where the original developer may no longer be around.

4. **Documentation agents** take a completed pipeline and a documentation template as inputs and produce compliant documentation automatically. This is critical for regulated industries in accounting, finance, and pharmaceuticals, where audit requirements make manual documentation a significant time sink. Generating a document is easy. Generating one that meets regulatory specifications every time is not - and that's exactly where most LLM-powered solutions fall short.



5. **Harmonization agents** onboard data from a variety of sources and convert it into a single harmonized format to process it in a unified way. Once the AI knows the harmonized format and has example transformation workflows, the agent can generate new workflows specific to new data sources as they are added.



The cumulative effect of these agents working together is not a marginal productivity boost. **It is a structural change in who can do data work, how fast it gets done, and how reliably it reaches production.**

The Productivity Math

The productivity improvements from this architecture compound across multiple dimensions, each of which independently reduces time to delivery:

Driver	Current State	With AI-native Data Prep
Pipeline first draft	Days to weeks	Minutes to hours
Validation loop	Sequential hand-offs between teams and iteration	Single shared artifact
Promotion to production	Engineer rewrites pipeline as code	Git review; visual workflow is same as code
Documentation	Written manually after the fact	Generated automatically, conforming to a template, always in sync
Error correction	Manual debugging, original developer required	Agent-assisted, actionable without deep context

The competitive framing matters: **the organizations capturing these gains first will operate at structurally lower cost.** Within 18 to 24 months, a peer organization in your industry will likely publish a case study describing how they deliver the same analytical output with significantly fewer resources. The question is whether your organization is the one publishing that case study or responding to it.

Key Takeaway

AI agents will not replace data analysts, but analysts who use AI will replace those who don't, creating a 30% to 50% productivity advantage that makes immediate, outcome-driven adoption a strategic imperative.

How to Adopt Without Disruption

The right adoption path is not a one-shot migration. It is coexistence followed by phased modernization, and the first phase can start immediately.

Phase 1: Designate an early adopter team (3-6 months):

Choose one team and one project, preferably a greenfield workload. The goal is to deliver a real project successfully on the new platform. Alternatively, existing Alteryx pipelines can be imported and converted to accelerate onboarding.

The first team will develop your organization's AI-first data development practices: how to prompt agents effectively, how to structure the inspect-and-refine loop, and how to establish Git-based governance for pipeline promotion. These practices become the template for every subsequent team.

Phase 2: Plan phased modernization (beyond 6 months):

Once you have internal evidence of real cycle times, real analyst productivity, and real promotion-to-production reliability, each subsequent team can scope and plan their own migration. The coexistence period is not wasted time; it is evidence accumulation that de-risks broader adoption.

What to avoid:

- Running a year-long migration project as the first step, which delays value and creates unnecessary organizational risk
- Waiting for the legacy vendor's AI roadmap to materialize, since architectural constraints mean the integration will remain shallow
- Moving slowly enough that competitors publish productivity gains before you do

What Successful Data Leaders Are Doing

The consensus view among technology leaders is not that AI agents will replace data analysts. It is expected that analysts who work with AI agents will replace those who don't. The dynamic mirrors what has played out in software development: the individual productivity of an AI-augmented developer is 1.3 to 1.5 times that of a non-augmented developer, and that gap is widening.

For data and technology leaders, the strategic imperative is clear:

Move early, move in a contained way, and measure outcomes. The cost of a failed pilot is low. The cost of watching competitors capture a 30 to 50% productivity advantage while waiting for certainty is high.

The organizations that will look back on this period as a competitive inflection point are the ones that identified a first use case, ran it to success, and used that internal proof to accelerate adoption. They are not the ones that completed a thorough vendor evaluation, while in the same quarter, their fastest-moving competitor went live.

What's Next for Your Data Prep?

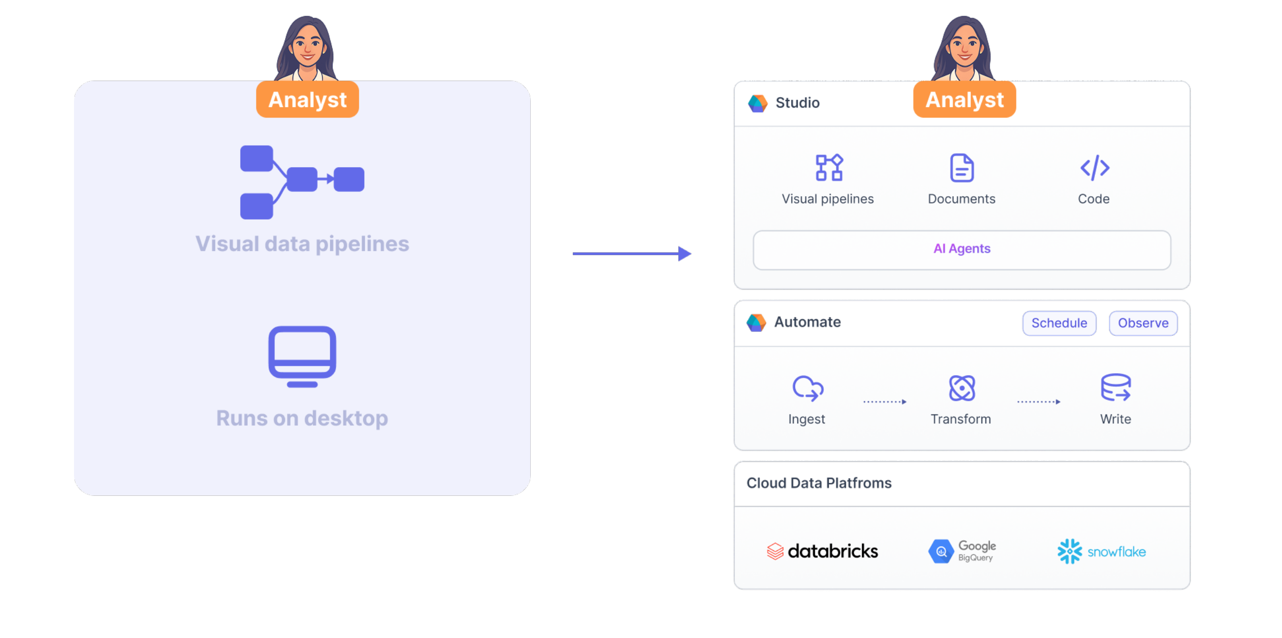
It's time to find your low-risk adoption path to enabling data analysts, skip the data engineering queues, and build modern workflows with AI.

Prophecy is the AI-accelerated data preparation and analysis platform that helps data analysts capture the productivity gains of AI for data work. Specialized AI agents (built on Claude Code) generate visual data workflows that users can inspect and refine; validated workflows compile to native SQL or Spark and deploy directly on Databricks, Snowflake, or BigQuery with full governance.

Prophecy provides an **Enterprise Express** program designed to take organizations from evaluation to delivered value within 90 days. It includes a structured success framework for identifying the right first team and workload and hands-on onboarding support from Prophecy's team.

With Prophecy, organizations are generating open source data pipelines. There is no lock-in. Customers choose to use Prophecy because of the value the platform brings, without imposing constraints.

Moving from legacy desktop tools to Prophecy



Where should you start?

Begin with a pilot project with a real team, a real deliverable, and a clear success criterion. Everything else follows from there. Get in touch with the Prophecy team today to modernize your data stack with AI.

Prophecy