8 Essential Tactics to fix what your data strategy is missing

The operational practices that turn a data-readiness plan into something that holds up in production.

svitla Global Digital Solutions Company

Intro

Strategy And Governance Get You To The Starting Line

Auditing your data, aligning it to a use case, and building a governance framework gets you to the starting line. It doesn't tell you what happens when two teams disagree on what a field means, when a fraud model has almost no real examples of fraud to learn from, or when someone uploads a poisoned dataset to a public model hub your team pulled from.

Here are 8 tactics that sit one layer below strategy and governance: the operational and technical practices that turn a data-readiness plan into something that holds up in production. Roughly in order of dependency: contracts and ownership come first, feature and pipeline infrastructure next, and the monitoring and security practices run continuously once the rest is in place.

1. Data contracts with a named owner
2. Synthetic data for rare-event and privacy gaps
3. Treating data as a security asset
4. Golden evaluation sets for agentic and LLM systems
5. Remove training skew with a feature store
6. Dedicated pipeline for unstructured data and RAG
7. Move to continuous data observability
8. Version and certify datasets like software releases

1

Put Data Contracts Between Every Producer And Consumer And Name An Owner

A data contract is a formal, versioned agreement that specifies schema, quality thresholds, and freshness for a dataset, so the team producing it and the team consuming it work from the same definition of "correct." Every contract needs a named, accountable owner on the hook for it, the same way a product manager owns a feature. Without either piece, small upstream changes break AI pipelines silently, and nobody finds out until the model is already wrong.

$12.9M

the average annual cost organizations report from poor data quality, according to Gartner's Magic Quadrant for Data Quality Solutions research.

2

Kill Training-serving Skew With A Feature Store

A feature store guarantees that the exact same feature logic runs in training and in production, so a "30-day average" means the same thing offline and online. Without it, subtle mismatches between training and serving pipelines quietly degrade model accuracy in ways that look like a model problem but are really a plumbing problem.

3

Use Synthetic Data To Fill Rare-event And Privacy Gaps Carefully

Fraud, equipment failure, and clinical deterioration are exactly the events your model needs to learn the most from and exactly the events your historical data has the fewest examples of. Generated data lets you manufacture realistic variants of the long tail without waiting years to collect it, and it sidesteps privacy restrictions on real customer records.

Gartner projected in 2021 that 60% of AI development data would be synthetic by 2024, a target date that has already passed, and adoption has been slower and messier than that forecast implied. The catch: synthetic data inherits and can amplify whatever bias or gaps exist in the real data it's generated from, so it needs the same scrutiny as the source data, not less.

Note: Example: a fraud team facing too few real fraud cases (tactic 3) and a feature store gap between training and live scoring (tactic 2) at the same time will often see a model that backtests well but misses obvious fraud patterns live, for two compounding reasons instead of one.

4

Use Synthetic Data To Fill Rare-event And Privacy Gaps Carefully

If any part of your AI strategy involves an internal copilot or a retrieval-augmented system, your documents, tickets, and transcripts need their own pipeline, one built around chunking strategy, embedding quality, and metadata-aware retrieval, not just schema and null checks. A model that retrieves the wrong three paragraphs is going to hallucinate no matter how good it is.

63%

of enterprise AI deployments now use retrieval-augmented generation, making unstructured-data readiness a mainstream requirement rather than an edge case.

5

Treat Training Data As A Security Asset

Public model hubs, crowdsourced labeling platforms, and scraped web data are all viable entry points for someone to quietly poison what your model learns. Apply the same scrutiny to training data that you'd apply to production code.

Key areas of focus should include:

- **Verified provenance for every external dataset or pretrained model before it's trusted**
- **Role-based access to training pipelines, not open access by default**
- **Validation gates before any external dataset ships into a training run**

6

Move From One-time Cleanup To Continuous Data Observability

Leading teams are shifting away from periodic cleansing projects toward always-on monitoring: automated anomaly detection, lineage tracking, and quality metrics that catch a broken upstream feed the same day it breaks, not the same quarter.

7

Build Golden Evaluation Sets For Agentic And LLM Systems

Traditional models get validated against a test set once. Agentic and LLM-based systems need a living evaluation set: curated questions and documents with known-correct answers that you run after every pipeline or prompt change to catch regressions in retrieval accuracy before customers do.

> Note:
>
> **A retrieval-precision score around 0.80 is a common threshold teams use before deciding a RAG pipeline is stable enough to ship.**

8

Version And Certify Datasets Like Software Releases

Ad hoc updates to a shared dataset are how silent breakage happens. Apply the same discipline you'd apply to a code release: version every dataset, run it through automated quality gates before it ships, and flag certified, production-ready datasets clearly in your catalog so teams aren't guessing which version is safe to build on.

The pattern underneath all 8: each of these practices exists because someone treated a piece of data infrastructure as a product, with an owner, a contract, and a version, and exposed it the way a platform team exposes an API, rather than treating it as a one-off report or a byproduct of some other process.

Ready For The Operational Layer?

Strategy and governance get your AI initiative funded. Data contracts, feature stores, and continuous observability are what keeps it running once it's live.

Svitla Systems helps teams build the technical layer underneath their AI strategy, from data contracts and feature infrastructure to RAG-ready pipelines and production monitoring.

Contact us today to start building a smarter, more efficient future for your business.

Learn More →

About Us

Svitla Systems is a global digital solutions company founded in 2003 and headquartered in Corte Madera, California, with delivery centers across the US, Latin America, Europe, India, and Asia-Pacific. Its teams of consultants, strategists, and engineers work across cloud, data and analytics, and AI/ML to help enterprise clients, from Silicon Valley startups to Fortune 500 companies, turn data foundations into production-ready AI systems.

The company holds ISO 9001 and SOC 2 certifications, is an AWS Advanced Consulting Partner, and is WBENC-certified as a women-owned business.

svitla Global Digital Solutions Company

HQ OFFICE
Corte Madera, CA, USA
+1 415 891 8605