

RAPTORS CONFERENCE 2026

Building a Production-Grade Enterprise Analytics Platform

Engineering Lessons from Scaling Business Intelligence at ADP

Janardhana Naidu Kola

Director of Business Intelligence
21 October 2026 · Online

DATA

GOVERN

SERVE

DECIDE

The production problem

- Dashboards are the visible layer. The harder work is underneath.
- Enterprise data arrives from systems built for different processes, owners, definitions, and timelines.
- Business rules change while the platform is running.
- At scale, trust, governance, performance, and recoverability become engineering requirements - not optional controls.

What changes when analytics becomes production infrastructure

FROM REPORTS

- Point solutions
- Manual reconciliation
- Local definitions
- Reactive fixes

TO A PLATFORM

- Reusable data products
- Shared business rules
- Controlled pipelines
- Operational ownership

THE REAL TEST

- Can it be trusted?
- Can it recover?
- Can it scale?
- Can it change safely?

Production grade is not a technology choice. It is a set of engineering behaviors.

A practical enterprise analytics architecture

1 Source systems

CRM · operational · financial · partner / carrier · reference data

2 Integration

ETL / ELT · standardization · identity matching · incremental processing

3 Governed data layer

conformed definitions · reconciliation · quality controls · lineage

4 Analytics & semantic layer

business logic · reusable metrics · dimensional models · aggregates

5 Consumption

dashboards · forecasting · operational analytics · leadership decisions

Lesson 1: integration is a business-logic problem

- The difficult joins are rarely only technical joins - they encode how the business actually operates.
- Keys can be missing, duplicated, delayed, or valid only within one source system.
- Build explicit matching and fallback rules instead of hiding exceptions inside reports.
- Separate source-system facts from derived business logic so changes can be traced and tested.

Design principle: make reconciliation visible before making the dashboard beautiful.

Lesson 2: data quality must be engineered into the flow

VALIDATE

Schema & type checks
Completeness
Allowed values
Freshness

RECONCILE

Cross-system totals
Lifecycle checks
Exception queues
Control totals

OBSERVE

Pipeline health
Refresh status
Volume shifts
Failure patterns

OWN

Named owners
Escalation
Root cause
Prevention

A trusted metric is the end result of controls, ownership, and repeatable validation.

Lesson 3: optimize the path, not just the query

- Reduce repeated transformation by moving stable logic upstream.
- Use incremental processing where full reloads do not add value.
- Pre-compute expensive, reusable calculations at the right grain.
- Design models around actual consumption patterns, not theoretical flexibility.
- Treat refresh windows, concurrency, and failure recovery as part of performance.

Lesson 4: governance should accelerate delivery

DEFINE

- Common metric definitions
- Business ownership
- Data contracts
- Change rules

CONTROL

- Access by role
- Certified datasets
- Release discipline
- Auditability

COMMUNICATE

- Known limitations
- Refresh expectations
- Change notices
- Decision context

Good governance removes ambiguity. Bad governance adds gates without reducing risk.

Lesson 5: stakeholder trust is a technical outcome

- Users trust analytics when definitions are consistent and exceptions can be explained.
- A fast dashboard with unclear lineage can be less useful than a slower, auditable one.
- Expose the reason behind important numbers: source, timing, logic, and known limitations.
- Build feedback loops with business stakeholders before discrepancies become credibility problems.

Where technical debt usually enters

Speed over reuse

Duplicated logic spreads across reports.

Flexibility over standards

Every team defines the same metric differently.

Patch over root cause

Exceptions accumulate until nobody knows the intended rule.

Scale over observability

More pipelines create more invisible failure modes.

Automation over ownership

A job can run successfully and still produce the wrong business result.

A production-grade operating model

BUILD

Versioned logic
Reusable patterns
Test before release

RUN

Monitor freshness
Handle failures
Meet service expectations

GOVERN

Own definitions
Control access
Document lineage

EVOLVE

Measure usage
Retire debt
Change safely

The platform is a product with a lifecycle - not a collection of finished dashboards.

Five takeaways

- Start with business processes and decision needs, then design the data model.
- Make data quality and reconciliation first-class engineering capabilities.
- Optimize end-to-end flow: ingestion, transformation, storage, semantic logic, and consumption.
- Use governance to create clarity, ownership, and safe change.
- Measure success by sustained trust and decision usefulness - not by the number of dashboards.

Production grade means engineering for trust.

Questions & discussion

Janardhana Naidu Kola
Director of Business Intelligence