DataDoctrine

60-Minute Data Stack Triage Checklist

Spot the pain. Stabilize the stack. No new tools required.

Overview

This checklist helps you assess your end-to-end data stack in under an hour. It’s designed for fast, high-leverage triage. No tool replacements, just clarity on what’s broken and what needs attention now.

Section 1: Surface Symptoms (10 Minutes)

Check all that apply. If you check 2 or more, your stack likely needs stabilization.

- [ ] Reports break or deliver incorrect results at least once a week
- [ ] Metrics differ between tools, dashboards, or teams
- [ ] You can't trace bad data back to its source
- [ ] Data cleanup consumes more time than analysis
- [ ] Business users frequently raise concerns about data trust

Section 2: Core Stack Components (30 Minutes)

Walk through each layer of your stack. Identify gaps, inconsistencies, and missing ownership.

Source Systems

- [ ] Are all source systems actively feeding your stack?
- [ ] Are there alerts or logs when source feeds break?
- [ ] Are source formats versioned or subject to silent change?

Ingestion (ETL/ELT)

- [ ] Are all pipelines running on schedule?
- [ ] Do failures trigger alerts or auto-retries?
- [ ] Are dependencies documented and testable?

Storage (Database or Warehouse)

- [ ] Are key tables and columns clearly documented?
- [ ] Are there known issues with missing or null-critical fields?
- [ ] Is schema change management in place?

Transformation Logic

- [ ] Are core metrics defined and reused (not re-coded)?
- [ ] Are business rules version-controlled and peer-reviewed?
- [ ] Is there duplicated or conflicting logic across tools?

BI and Reporting Layer

- [ ] Do dashboards use consistent metric definitions?
- [ ] Are filters and joins clearly documented?
- [ ] Are users trained on how to interpret reports?

Section 3: Ownership and Access (10 Minutes)

- [ ] Does each layer have a clearly defined owner?
- [ ] Are data consumers able to request access without workarounds?
- [ ] Are production, test, and dev environments separated?
- [ ] Is there role-based access control at the warehouse and BI levels?
- [ ] Are changes to logic or pipelines reviewed and approved?

Optional: Triage Scoring Table

Use this table to summarize what you found and prioritize fixes.

Area	**Risk Score (1 to 5)**	**Notes**
Source System Reliability		
Pipeline Stability		
Data Model Clarity		
Metric Consistency		
Dashboard Trust		
Ownership and Governance		

What to Do Next

This checklist is not your solution, it’s your snapshot. Use it to:

- Prioritize root-cause fixes over patchwork
- Document what’s owned and what’s not
- Start conversations with your team based on facts, not feelings