

The Migration Exposure Test

Trusted BI. Prove every number you publish.

Run the checklist against your own environment, not against a vendor's feature list. Work through it with the two or three people who would actually have to answer, and time the answers where noted. The time it takes to answer is usually the finding.

Score at the end of each one. The score tells you whether the problem is a workflow annoyance or an architecture gap, and the two need different budgets.

Migrations fail in two ways. The team doesn't know what depends on what, so a schema change cascades silently into production. Or the team rebuilds years of BI semantic models by hand in the new platform's format, slowly and slightly wrong.

Both are avoidable before the first query moves. Both are usually discovered after.

Section A: Size the exposure

- ☐ Source platform and target platform: [FILL IN]
- ☐ Announced or expected cutover date: [FILL IN]
- ☐ Count of semantic models, data sources, or reports in the source platform: [FILL IN]
- ☐ Years of accumulated definition work represented there: [FILL IN]
- ☐ Is anyone currently planning to rebuild those definitions by hand in the target? Yes / No
- ☐ Do you have a complete inventory of downstream consumers for the tables involved? Yes / No

Section B: Time the answers

- ☐ Time to list every downstream consumer of one migrating table, column by column: [FILL IN]
- ☐ Time to answer “what breaks if this column is renamed”: [FILL IN]
- ☐ Estimated engineer-weeks budgeted for definition rebuild: [FILL IN]
- ☐ Estimated engineer-weeks budgeted for post-cutover reconciliation: [FILL IN]

Section C: Requirements to test on any vendor call

- ☐ Can the platform read your existing Power BI semantic models, Tableau data sources, and LookML definitions and import them as governed starting points? Semantic Hub's reverse-engineering engine does this, so years of existing semantic work become the foundation for the new platform instead of a rebuild.
- ☐ Does impact analysis run at column level across the full estate, including systems whose jobs haven't run recently?
- ☐ Is that impact analysis available before the change deploys rather than after something breaks?
- ☐ Once imported and governed, can a definition compile into both the old and the new platform at once, so the two agree during the cutover window?
- ☐ Remove the vendor a year after cutover. Do your definitions stay? Compiled artifacts live inside your platforms, so they should.
- ☐ Ask for the reverse-engineering demo against your actual models in the first session. A prepared dataset proves nothing about your definitions.

Score

1 point each for: a rebuild-by-hand plan in place, no complete downstream inventory, over an hour on the downstream-consumer timer, more than 4 engineer-weeks budgeted for rebuild, a cutover date inside 6 months.

0 to 1. The migration is scoped. Keep the inventory current.

2 to 3. You're carrying schedule risk that hasn't been priced. The reconciliation work usually lands after the cutover date, when the team has already moved on.

4 to 5. The migration is exposed on both failure modes at once. The definitions and the dependency map both need to exist before anything moves.

Next step

Take the rebuild estimate from Section B and put it next to the cost of importing those definitions once and compiling them into both platforms. That comparison is the business case, and it survives contact with a CFO.

Get in touch.

Request a demo to learn more about MetaKarta and how it helps you reduce database and BI platform migration risk.

[Request Demo](#)