Qualibri Tech

✳ Case study

The Unit Tests Passed. The Dashboards Didn't.

How Qualibri Tech replaced fragmented testing on a growing analytics platform with one risk-based data quality strategy, a code-based automation pilot, and a foundation ready for future regulated validation.

Client	→	A European pharma company running a clinical-development analytics platform
Stack	→	Snowflake × dbt × Tableau × SQL × Python × Airflow × REST APIs
Challenge	→	A platform growing faster than its testing — the warehouse had unit tests, the reporting layer had almost none, and each team tested in isolation
Solution	→	A data quality risk assessment, a unified risk-based test strategy, defect / release / UAT processes, and a code-based automation pilot
Results	→	**One testing strategy across teams × fewer production defects × a maintainable automation pilot × a foundation ready for regulated validation**

Client profile

The client is a European pharmaceutical company whose clinical-development analytics platform turns data from external and internal sources into Tableau reports. Its business users rely on them to decide what to do next. Unlike a research lab, it does not run its own studies — it consumes data others produce and analyzes it. The decisions it supports, and the patients affected downstream, depend on that data being right.

The platform was growing fast — more analytical products, integrations, and users — across a Snowflake and dbt warehouse and a reporting layer where a custom API feeds Tableau dashboards. It was also preparing for GxP validation, the regulated standard for computerized systems in life sciences, which demands documented, repeatable quality processes.

The challenge

When Qualibri joined, data testing had not kept up with the platform. Defects were regularly found in production by real users: a dashboard that failed to load or a report showing half-missing data. Many were the kind a simple smoke test would have caught. Digging into a complaint often revealed business logic implemented incorrectly, or never verified.

The deeper problem was organizational. Each team owned a part of the platform and tested — when it tested — in isolation. There was no shared testing strategy, shared definition of data quality, or picture of how the pieces fit together. Changes to upstream sources — 4 to 5 feeds — arrived with no warning, so the team learned about breakages in production.

The ask: make testing something the whole platform could rely on, and get it ready for validation.

The solution

The engagement was advisory: diagnose first, then build the strategy and processes to act on it.

First, a shared picture. A structured data quality risk assessment mapped testing maturity, defect handling, releases, automation opportunities, documentation, and the gaps between teams. It found that unit tests existed, but no unified strategy or shared standard of data quality covered the platform as a whole.

Then, a strategy every team could share. Qualibri defined a risk-based test strategy and the processes to support it: a structured defect lifecycle, clear release-readiness criteria with go / no-go decisions, and a formal User Acceptance Testing (UAT) process for real end users, built in Jira with runs, cycles, and coverage reporting. It also showed the warehouse team how to replace dozens of hard-coded, single-column dbt tests with reusable, parameterized ones.

Then, automation on the existing stack. For the untested reporting layer, a code-based pilot in Python — Playwright on the Tableau dashboards, plus REST-API checks — grew from 15 smoke checks to ~25 automated tests that load each critical dashboard, confirm nothing failed, and reconcile figures against the latest data load. It was the reporting layer's first automated coverage.

Finally, a foundation for validation. New testing processes were shaped toward what GxP would later require: improved documentation, traceability from requirement to test, repeatable testing, and corrective and preventive actions (CAPA) to stop recurring source-data problems at the root.

Results & metrics

Before	After
Each team tested in isolation, with no shared strategy	**One risk-based test strategy and a shared definition of data quality across teams**
The reporting layer had almost no testing	**A code-based Python + Playwright pilot — ~25 tests, the layer's first automated coverage**
Defects found by users in production	**A structured defect lifecycle and release-readiness criteria with go / no-go decisions**
Delivered reports sometimes didn't match what users expected	**A formal UAT process in Jira, with runs, cycles, and coverage reporting**
Source-data changes broke reports without warning	**CAPA practices to catch and prevent recurring data issues upstream**
Processes not ready for regulated validation	**Documentation, traceability, and repeatable testing toward future GxP validation**

Other results:

- Fewer production defects on the reporting layer, and earlier detection of the rest.
- Business logic and analytical figures validated with more confidence.
- Stronger collaboration between the development, business, and QA teams that had worked in isolation.

Not sure what your current tests actually catch?

We suggest starting with a focused data quality risk assessment. Book a 30-minute scoping call, and we'll tell you which flows we'd map first and why — and honestly, whether an assessment is worth it for your platform.

Contact me to book the scoping call

Natalia Zhornyk

Founder
at Qualibri Tech