CASE STUDY

"The Lab Shouldn't Wait on Engineering"

How NonStop Gave Clinical Labs Full Control Over Their Reporting

Clinical Genomics · Reporting Infrastructure · Lab Autonomy

Executive Summary

Top 5 Problems

1. Hardcoded reporting — every change needed a dev cycle
2. No audit trail for template changes — accreditation risk
3. Clinical staff writing Jira tickets instead of owning workflows
4. Manual sign-off on every report — fatigue and error risk at scale
5. Every new panel or client customization was a fresh engineering project

Impacts Those Problems Created

1. New report types took 3–4 months to go live — TAT commitments broken
2. CAP/CLIA/ISO 15189 compliance exposed — no versioning, no rollback
3. Clinical expertise trapped in a ticket queue — talent misallocated
4. High-volume routine panels created reviewer fatigue and error surface
5. No reuse, no compounding — every new panel cost as much as the first

Solution We Built:
Role-separated Report Builder + Template Versioning + Decoupled Report Service + Automated Sign-Off Engine

Impact of the Solution

- New report types: 3–4 months → 1–2 weeks
- Reporting changes: sprints → hours
- Developer dependency: eliminated
- Manual sign-off load: significantly reduced via auto-approval rules
- Client customization TAT: weeks → days
- Accreditation readiness: full version history, clean rollback on every template

The Real Cost of a Developer-Dependent Reporting System

For a clinical genetics lab director, the report isn't an output. It's the product. Every result that reaches a clinician or patient travels through that document, and when the system producing it is slow, rigid, and dependent on an engineering queue, the consequences reach far beyond inconvenience.

Our client was living that reality. Every report layout, every field, every structural change was hardcoded. Routine updates required a full development cycle. New report types took three to four months from request to delivery. Lab technicians, the people who understood best what a report needed to communicate, had no direct way to act on that knowledge.

> **"Lab technicians, the people who understood best what a report needed to communicate, had no direct way to act on that knowledge. They were writing tickets and waiting."**

What a Lab Director Was Actually Dealing With

Regulatory Exposure Without an Audit Trail

CAP, CLIA, and ISO 15189, lab directors operate under accreditation frameworks that demand traceability. Every report format change needs to be documented, versioned, and reversible. With a hardcoded system, even a minor field label update left no clean trail.

Broken TAT Commitments

Lab directors make turnaround time promises to hospital clients and health systems. When a new test panel was required, three to four months of reporting development were required before it could go live.

Clinical Talent Writing Jira Tickets

Experienced lab technicians submitting formatting requests to an engineering queue is more than an inefficiency; it's a signal. It tells skilled clinical staff that their judgment doesn't translate to agency in their own workflows.

Fatigue-Driven Error Risk at Scale

High-volume routine panels, carrier screening, and pharmacogenomics generate hundreds of reports daily. Manual sign-off on each one isn't just slow; it's where reviewer fatigue creates error risk.

Multi-Panel Complexity With No Scalability Path

A modern diagnostics lab doesn't run one report type. Oncology panels, hereditary disease, prenatal screening, and pharmacogenomics each have different fields, layouts, and sign-off logic. Every new panel was a fresh development project. There was no compounding, no reuse.

Client Customisation Requests Becoming Engineering Projects

Health system clients want reports formatted to their standards, their branding, their field order, and their clinical terminology. Pre-platform, every customization request entered the same development queue as infrastructure work.

What We Built and Why

The diagnosis was clear: the bottleneck wasn't developer speed. It was that content and configuration decisions were being treated as software problems. The fix wasn't a faster development cycle; it was removing development from the loop entirely for routine reporting decisions.

We structured the platform in four layers, each addressing a specific failure point.

Separation of Roles at the Surface

Two distinct interfaces serve two distinct workflows. The Report Builder is a designer-facing workspace where lab teams compose templates, define fields, and publish new report structures without touching a development environment. The Report UI is where end-user workflow lives: pulling up a generated report, reviewing it, and signing off. Role-based access ensures the right people stay on the right surfaces.

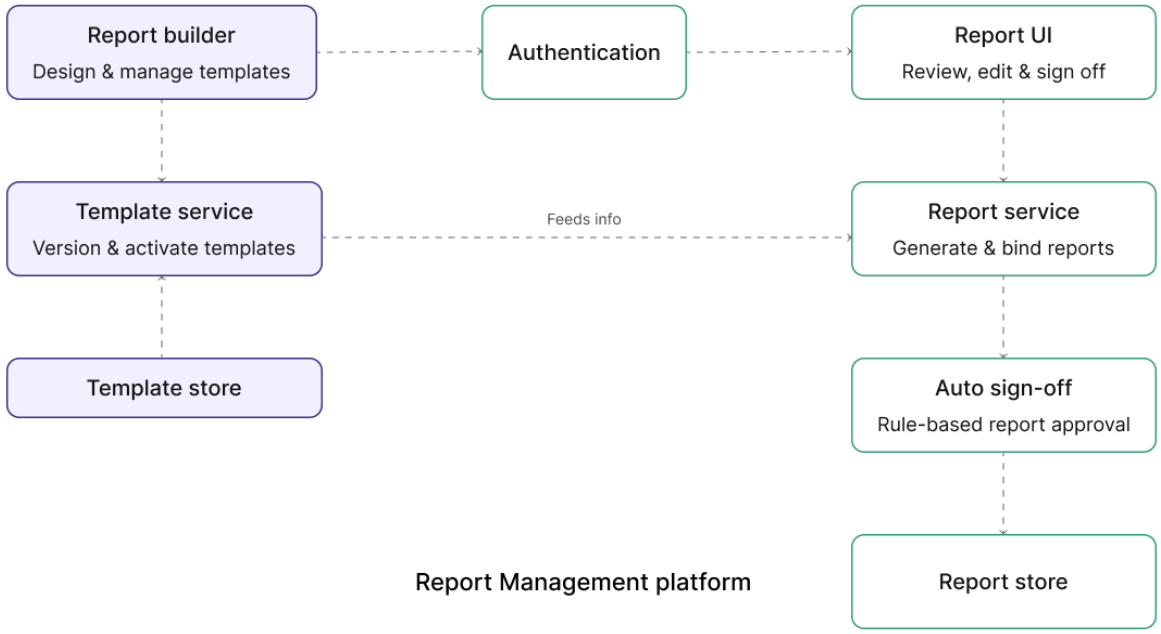
> **"The design principle: lab teams should own their reports. Developers should be building platform capabilities, not formatting fields."**

Template Versioning for Accreditation Confidence

The Template Service manages the full lifecycle of every report template, from creation to versioning and activation. Lab teams can iterate on a template in draft without affecting reports already in flight, and roll back cleanly if something isn't right. This isn't just operational safety. It's the audit trail that accreditation frameworks require.

Decoupled Services for Independent Evolution

Template management and report generation run as two independent services. The Report Service fetches the active template version, binds it dynamically with lab data, and produces a report ready for review. The dependency is one-directional: templates feed report generation, never the reverse.



Automated Sign-Off for Routine Volume

The Auto Sign-Off Processor evaluates completed reports against a configurable rule set and approves them automatically when criteria are met. For high-volume, routine report types, this removes the manual review step entirely, redirecting expert attention to the cases that actually warrant it.

Separated Data Domains for Scale and Stability

Template data and report data live in independent stores, mirroring the service architecture. Each domain can be queried, backed up, and scaled without affecting the other. Report generation operations never degrade the template authoring experience.

By the Numbers

Metric	Before	After
New report type delivery	3-4 months	**1-2 weeks**
Reporting change cycle	Sprints (weeks)	**Hours**
Developer dependency	High	**Eliminated**
Manual sign-off load	100%	**Significantly reduced**
Client customisation TAT	4-5 Weeks	**In days**
Template reuse across panels	None	**Compounding**
Accreditation audit trail	Absent	**Full version history**

The Impact

- **Turnaround dropped from months to days.** Commercial launches no longer wait on engineering, new report types that previously required 3-4 months now go live in days.
- **Lab teams became self-sufficient.** Day-to-day reporting decisions moved entirely into lab team hands. Engineering time freed up for platform capability work.
- **Accreditation risk reduced.** Template versioning gave lab directors a clean, auditable change history. Every modification is traceable, every rollback is clean.
- **Manual review load cut significantly.** Automated sign-off on routine report types eliminated the high-volume review queue, reducing fatigue risk and redirecting clinical attention where it belongs.
- **Reusability compounded over time.** Component reuse across report types meant each subsequent panel cost a fraction of the first to build. Efficiency gains accelerated as the template library grew.
- **Client customisation became a lab task.** Health system-specific formatting requests moved from the development queue to the Report Builder. What previously took weeks now takes hours.

> **“The lab team went from being dependent on engineering for every reporting change to being fully autonomous on day-to-day reporting decisions, which is where clinical knowledge belongs.”**

A lab director’s job is clinical leadership, owning test quality, managing accreditation, building health system relationships. It shouldn’t include managing a reporting change queue.

NonStop built a platform that put reporting ownership exactly where clinical knowledge lives: with the lab team. The result wasn’t just faster reports. It was a lab that could move at the speed its clinical work demanded.

Interested in building the same capabilities for your lab?

nonstopio.com | Clinical Genomics & Life Sciences Engineering