

Laboratory Informatics: Five Kinds of Data, Three Decisions

The term names five different kinds of laboratory data at once, which is why it feels vague. Separate them, decide who owns each, and the shortlist writes itself.

Brandon Holland, Marketing & Operations Digital Solutions Architect

July 13, 2026 · 11 min read

IN BRIEF

Laboratory informatics is not a product category. It is the management of five kinds of laboratory data: sample identity, experiment reasoning, instrument output, reportable results, and evidence the work happened. Different systems own different ones, which is why most laboratories need two or three capabilities rather than one platform.

Key takeaways

- Laboratory informatics names five kinds of data at once: sample, experiment, instrument, reportable result, evidence.
- Nothing owns the evidence of activity by default, which is why that is where assessment findings are made.
- Cost the current process from four records you already hold; a borrowed industry percentage loses the room.
- Audit trails buy a permanent review obligation. The missed review is what gets cited, not the trail.
- Three questions decide fit: who owns sample identity, what must be reconstructable how fast, and which seam breaks your week.

Laboratory informatics is not a product you can shortlist. It is the set of decisions about where each kind of laboratory data lives and what has to be rebuildable from it, and until those decisions are made no product can be evaluated against anything.

<https://www.lablynx.com/resources/articles/laboratory-informatics/>

Most laboratories asking the question need two or three specific capabilities rather than a platform, and the fastest way to find out which is to look at where your data currently falls between systems. This covers the five kinds of data the term actually names, which software owns each, what the whole thing returns and what it will not fix, how to cost your current process from records you already hold, and the three questions that make a shortlist possible.

What is laboratory informatics?

Laboratory informatics is the management of five different kinds of laboratory data. Each behaves differently, and each has to be owned by some system.

Sample data. The identity of a physical thing that arrived, was split, was tested and was reported. It has to survive aliquoting and relabelling, or a report cannot be traced back to a tube.

Experiment data. Why a piece of work was done the way it was: the reasoning, the deviation, the thing that was tried and abandoned. It has no fixed shape, which is why it resists the fields a sample system wants to put it in.

Instrument data. The raw output and the processing applied to it. This is where the record of what was done to a number lives, and it usually sits on the instrument rather than anywhere central.

Reportable results. The values that leave the building, carrying the qualifiers, uncertainty and format the recipient requires. In environmental work that is a named state deliverable at a version number, not a PDF.

Evidence of activity. The records showing the work happened as described, covering preparation, calibration, competence and review. It is what an assessor asks for, and it is the one nobody buys software for.

The word feels vague because it names all five at once. Naming which of the five your laboratory has no owner for is the work that makes a shortlist possible, and it is work no vendor can do for you.

Which laboratory informatics software owns which kind of data?

Each product category was built around one of the five, and stretches badly when asked to hold another. This is the map, and the right-hand column is where most laboratories are actually bleeding.

KIND OF DATA	USUALLY OWNED BY	WHAT GOES WRONG WHEN NOTHING OWNS IT
Sample identity	LIMS	Two systems disagree about what a sample is
Experiment reasoning	Electronic lab notebook	Nobody can say why a method was chosen
Instrument output	Chromatography or scientific data system	The processing behind a number is unreviewable
Reportable result	LIMS, in the client's format	The deliverable is rebuilt by hand every time
Evidence of activity	Nothing, by default	The finding at your next assessment

A [LIMS](#) owns the first and part of the fourth. An [electronic lab notebook](#) owns the second. A chromatography data system owns the third, and [instrument integration](#) is what moves the reviewed number from there to the sample record without a person retyping it. Nothing owns the fifth by default, which is why it is where findings are made.

"Integrated laboratory informatics" means one thing in practice: a single sample identifier that crosses every one of those seams intact, so a result in the report can be walked back to the instrument run, the method version and the tube. It does not mean one vendor. The [laboratory software overview](#) covers the categories in more depth.

Why is the answer usually two or three things, not a platform?

Because the problem most laboratories describe is not an absence of software. It is the seams between the software they already have.

In LabLynx's own inbound inquiries, roughly half of laboratories making first contact report multiple departments or sites, and about a third already run a LIMS, LIS or a named laboratory product rather than

paper. Only about one in eight describes notebooks or paper when asked about its current state. So a replacement is usually a migration, carrying all the reconciliation that implies, rather than a first installation. That single fact changes what you should be asking a vendor to prove.

The other signal is what people volunteer when nobody prompts them. Asked to describe the need in their own words, they mention integration and the report they have to produce more often than they mention instruments. Asked directly whether they need instrument interfaces, most say yes. Both are true, and the difference matters: the thing that hurts enough to write about is the output and the connections, and that is a seams problem rather than a platform problem. Four wastewater plants consolidating sample login. An organisation with a research system and nothing on the clinical side. Two locations on separate platforms. Fragmented enough that no single system sees the work, and rarely resourced enough to have anyone whose job this is.

What does your current process already cost?

Most laboratory software business cases fail in the same place. They open with an industry-average savings percentage the reader has no reason to believe, and the conversation becomes an argument about whether that percentage applies to this lab. It usually does not. A number a CFO will accept is built the other direction: it starts with what your lab already spends, drawn from records you can pull yourself.

Four lines, each traceable to a document someone in your lab can produce. Use a fully loaded hourly rate throughout — salary plus benefits, taxes and overhead — and agree the multiplier with finance before you calculate, which keeps the later conversation off the arithmetic.

COST LINE	WHAT TO COUNT	FORMULA
Documentation and transcription labour	Hours per week retyping results between systems, assembling reports by hand, filling logs. Sample two or three ordinary weeks; estimates run low	weekly hours × loaded rate × 52
Repeat tests caused by documentation	Only repeats where the analysis was sound and the retest happened because a record was missing, unsigned, unlinked or could not be found in time	repeats per month × loaded cost per test × 12
Assessment and audit preparation	Person-hours preparing for the last assessment, including time spent locating records that existed but could not be found quickly	hours per event × loaded rate × events per year
Findings that recurred	Findings appearing in both of the last two assessment reports. A recurring finding was closed on paper, not in practice, and will be paid for again	hours to close × loaded rate, summed across recurring findings

The second line is what separates a serious case from a generic one. You are not counting every repeat, only the ones the analytical work did not cause. Your corrective-action and repeat-test logs show these once you filter for cause. The sum of the four is the honest annual cost of the current process, and it is defensible because every input traces to a record.

What does laboratory informatics return, and what will it not fix?

The return is that a named result can be reconstructed on demand, and that is worth paying for precisely because the alternative is somebody rebuilding it by hand under time pressure, usually during an assessment. No credible percentage exists as a general answer: the return depends on your documentation burden, repeat-test rate, assessment frequency and staff cost, all of which vary widely between labs of the same size. Any vendor quoting a universal figure is quoting a number they cannot support for your operation.

What it costs to keep is the part usually left out of the case. Turning on audit trails buys a permanent review obligation rather than a one-off configuration: FDA's data integrity guidance expects audit trails reviewed at the same frequency as the data they protect, and where no frequency is specified, the lab sets one by risk assessment. Somebody has to read them, and it is the missed review that gets cited, not the trail. Budgeting for the licence and not for the reviewing is how a system that passed its validation still produces findings two years later.

It will also not buy audit readiness. Ask an ISO/IEC 17025 assessor what they cite most often and the answer is rarely a missing electronic audit trail; it is incomplete management-review inputs, particularly the risk identification under clause 8.9.2. No system generates that for you.

Naming is the same shape of problem. Cross-site search and any analytics worth having require agreed names for every attribute you intend to collect, embedded in the configuration before go-live. Letting each department keep its own convention is fatal to anything you later want to ask across them, and a platform only industrialises the collisions. That work is a fortnight of arguments about vocabulary and it has to happen either way.

And a research group should probably not start here. If what your laboratory primarily produces is experiments and changing protocols rather than reportable results at volume, a notebook is the first system and a sample platform is the second; [LIMS vs ELN](#) covers that choice. Buying them in the other order produces an expensive filing cabinet.

What is the ROI timeline for implementing a modern LIMS?

Payback timing is a calculation, not a benchmark. Divide your first-year investment by the monthly portion of current-state cost your lab judges recoverable.

Payback period in months = first-year investment ÷ ((annual current-state cost × recoverable portion) ÷ 12)

First-year investment is licence plus implementation plus validation, split between vendor and your quality unit, plus data migration, plus internal staff time. The recoverable portion is a judgement your team makes and documents, not a number a vendor supplies, and stating it that way is what makes the timeline defensible. A case that says “we judge 60 percent of documentation labour recoverable, based on which tasks the system replaces” survives scrutiny. One that says “60 percent savings” does not. Be conservative and say so: a case that clears the bar at a cautious assumption invites approval, one that clears it only at an optimistic assumption invites negotiation over the assumption.

Two timing realities belong in the same paragraph. Implementation takes months, not weeks, so recovery does not begin at contract signature; and the first months after go-live are slower, not faster, while the old system runs in parallel and staff learn the new one. A payback figure that starts the clock at contract and assumes full recovery from day one is wrong in a knowable direction.

NOTE

The figure that appears on no quote: internal staff time. Your people will spend hours in requirements sessions, configuration review, user acceptance testing, SOP rewriting, data cleanup before migration and training. None of it appears on a vendor quote because the vendor is not selling it, and it is routinely the largest under-counted item in the case. Ask each vendor for the internal hours they have seen at labs of comparable size. If a vendor cannot answer, that is information about the vendor.

What should you work out before you shortlist anything?

Three questions decide fit, and none of them is a feature comparison.

Which system owns the sample identity? Exactly one should. If two do, you have a reconciliation problem that grows with volume.

What must be reconstructable, and how fast? An assessor asking for everything behind one named result within the hour is a different requirement from an annual report, and it is the requirement that actually shapes the architecture.

Which seams are load-bearing today? The connection that breaks your week is the one to fix first, and it is usually to a billing system, an electronic health record or a state deliverable rather than to an instrument.

Answer those three and the shortlist writes itself, because most products are clearly right or clearly wrong against them. Skip them and you will compare feature grids that all look adequate. Then make every quote comparable before you compare any of them: the differences between quotes are frequently differences in what was included rather than differences in price, and you cannot see that until the line items match. [Lab management consulting](#) exists for laboratories that want the three questions settled with someone who has done it before.

- One named system owns sample identity, and the second candidate has been told it does not
- The reconstruction requirement is written down with a time limit, not “full traceability”
- The load-bearing seam is named, and it is the first thing on the integration list
- Every quote states licence cost and the variable that scales it, at 150% of today’s volume
- Every quote prices implementation against a written scope, with what triggers a change order
- Every quote splits validation deliverables between the vendor and your quality unit
- Every quote states what is migrated, verified how, and what happens to records that do not map
- Every quote gives support response commitments by severity, and who assigns severity

What a LIMS carries in this work

Two of the five kinds of data, and the seams between them and the other three. A LIMS owns the sample identity and the reportable result, carries one identifier across every aliquot and instrument run, and produces the deliverable in the recipient’s format rather than a PDF somebody rebuilds. It does not own the experiment reasoning, and it does not own the raw instrument processing; it receives the reviewed number from the system that does. And it does not create governance: the fortnight of vocabulary arguments

happens before configuration or it happens, more expensively, after.

How LabLynx handles this

The LabLynx laboratory software owns the sample identity and the reportable result, and is built to make one named sample reconstructable quickly. If your load-bearing seam is a research notebook or a chromatography data system, those remain separate systems and should; we interface to them rather than pretending to replace them. If your laboratory has twelve people and no data dictionary, the first conversation we will have is about the dictionary, not the licence.

Frequently asked questions

What is the difference between laboratory informatics and a LIMS?

Laboratory informatics is the problem space; a LIMS is one product category inside it. Informatics covers sample identity, experiment reasoning, instrument output, reportable results and evidence of activity. A LIMS owns the first and most of the fourth. Notebooks, chromatography data systems and instrument interfaces own others, and the fifth is owned by nothing unless a laboratory decides it is.

Do we need a full laboratory informatics platform?

Usually not at first. The pattern in inbound inquiries is a laboratory spread across multiple sites or departments that already runs a named product, with pain concentrated at two or three seams: a deliverable rebuilt by hand, a billing or health-record interface, sample login duplicated across locations. Fixing those seams is a smaller, faster and more defensible project than replacing everything.

How do you calculate LIMS ROI?

From your own records, not a benchmark. Sum four current-state costs — documentation labour, repeat tests caused by documentation rather than analysis, assessment preparation, and findings that recurred across two assessments — at a fully loaded hourly rate. Then set first-year investment against the portion

your team judges recoverable, and document who made that judgement and why.

Where do most laboratories go wrong with informatics?

Naming. Cross-site search and analytics require agreed names for every attribute before configuration, and letting each department keep its own convention makes anything you later ask across them impossible. A platform does not fix that; it stores the same collisions in a more expensive place. The second failure is budgeting for the licence and not for the people who must review the audit trails it generates.

Should a research laboratory start with a LIMS or an ELN?

If what the laboratory primarily produces is experiments and changing protocols rather than reportable results at volume, a notebook is the first system and a sample platform the second. Buying them in the other order produces an expensive filing cabinet: a sample system with nothing to track, and researchers still writing their reasoning somewhere it cannot be found.

What should we decide before shortlisting products?

Three things. Which single system owns the sample identity, because two owners create a reconciliation problem that grows with volume. What must be reconstructable and how fast, because an hour and a year are different architectures. And which seam is load-bearing today, because that is the first integration to price, and it is usually a billing system or a state deliverable rather than an instrument.

Sources and references

1. [21 CFR 11.10 Controls for closed systems](#)

Electronic Code of Federal Regulations

2. [Data Integrity and Compliance With Drug CGMP: Questions and Answers, Guidance for Industry](#)

U.S. Food and Drug Administration

3. [ISO/IEC 17025:2017 General requirements for the competence of testing and calibration laboratories](#)

International Organization for Standardization

Brandon Holland

Marketing & Operations Digital Solutions Architect

Brandon Holland is LabLynx's Marketing & Operations Digital Solutions Architect. He builds and runs the digital systems behind the company's public surface, from the website and content architecture to the tooling that keeps product information accurate everywhere it appears. He writes about laboratory informatics from the systems side: how lab records get structured, searched, and kept defensible as a lab scales.

Read the current version online

<https://www.lablynx.com/resources/articles/laboratory-informatics/>

Downloaded September 24, 2026. LabLynx builds laboratory informatics software: LIMS, ELN, LIS and lab automation.