Too Big for Spreadsheets, Too Small for a LIMS Project

The lab in the middle has structure and no headcount for an implementation. Neither the spreadsheet nor the six-month project fits, and the category between them rarely gets named.

Brandon Holland, Marketing & Operations Digital Solutions Architect

August 20, 2026 · 8 min read

IN BRIEF

A lab that has outgrown spreadsheets but cannot justify a LIMS project needs off-the-shelf laboratory software: records with real fields and templates its own people write, a spreadsheet import that shows what it will do before writing anything, dashboards the team arranges, and a door for customers. Usable in days, not configured over months.

Key takeaways

- The middle lab's problem is not features. It is that every option on the market assumes a project team the lab does not have.
- A spreadsheet has been outgrown the moment it cannot be reviewed, signed or asked a question, whatever its size.
- Off the shelf means the lab's own people write the templates and change the forms, in minutes, without a vendor or a ticket.
- An import that shows what it will create, update and skip before it writes is the difference between migration and damage.
- Customers who can see their own work stop phoning the lab, and the lab's people get their afternoons back.

There is a laboratory that every vendor's website skips. It has twelve people, two hundred samples a month and a set of spreadsheets that have grown fields, tabs and colour codes over six years. It has a quality manager who knows the spreadsheets cannot be signed, an owner who has priced a LIMS implementation, and nobody whose job could be an implementation project for half a year.

That lab is in the middle, and the middle is not small. Most working laboratories are closer to it than to either the two-person startup the free tools are written for or the multi-site enterprise the LIMS brochures picture. This article is about what fits it: why the spreadsheet fails and why the project fails, what off-the-shelf laboratory software should do on its first day, how the spreadsheet should come with you, and what the lab's customers should see.

Which lab is the one in the middle?

It has structure. Samples arrive with a customer, a matrix and a set of tests; results are reviewed by someone other than the analyst; reports go out with a name on them. The structure is real and the spreadsheet encodes it, in column headings and conditional formatting that one person understands fully.

It has no headcount for a project. A LIMS implementation assumes a project team: someone to own requirements, someone to sit in configuration workshops, someone to run validation, someone to migrate the data. In the middle lab those are all the same person, and that person also signs the reports. The vendor's six-month plan is honest about the work and silent about who does it.

And it has a customer who wants to see things. Not a portal with a logo, just a way to find out whether their samples have arrived and when the results are due without telephoning. The spreadsheet cannot show them that. Neither can most LIMS quotes without another module and another seat.

The middle lab is often described as "not ready for a LIMS", which puts the problem on the lab. The problem is the other way round. Every option on the market assumes a project team the lab does not have, and the lab responds by keeping the spreadsheet for another year, which is what it did last year.

Why does the spreadsheet fail, and why does the LIMS project fail?

The spreadsheet fails when someone asks it a question it was not built to answer. It cannot be reviewed, because there is no record of who changed a cell or what it held before. It cannot be signed, because a signature on a printout is a signature on a moment, and the file has moved on. It cannot be asked which samples from one customer were tested by one analyst on one instrument last quarter without somebody building a pivot table and hoping the column names held still. When an accreditation assessor asks any of those questions, the spreadsheet has been outgrown, whatever its size.

The LIMS project fails differently. It does not fail on capability; the system can do everything the lab needs and a great deal it does not. It fails on time and attention. Configuration workshops need the lab's people in the room. Requirements need to be written by someone who knows the lab. Validation needs test scripts executed and signed. Data migration needs the spreadsheet's six years of quiet inconsistency resolved row by row. For a lab without a project team, the project either stalls at month four with the spreadsheet still in use, or finishes with a system configured to the vendor's assumptions because nobody from the lab had time to disagree.

> **NOTE**
>
> Neither failure is a failure of the people. A spreadsheet is the right tool for a lab of three and the wrong tool for a lab that has to be audited, and the transition between the two is not marked by anything visible. A LIMS project is the right shape for a lab that can staff it. The middle lab's mistake is believing those are the only two shapes.

What should off-the-shelf laboratory software do on day one?

Off the shelf means the lab's own people make it fit, in days, without a vendor project. In practice that comes down to a short list of things the software should do before anyone from the vendor is needed.

- Hold records with real fields, so a temperature is a number that can be filtered and charted, not a phrase buried in a note
- Let the lab write its own templates for the work it does more than once, and change them in minutes when the method changes
- Let two groups on the same installation work differently, with different words, forms and applications switched on per group
- Record who did what and when on every record, without anyone switching it on or assembling it afterwards
- Bring the existing spreadsheet in, checked, with nothing written until the person importing it has seen what will happen
- Let the team arrange its own dashboard from the records it already keeps, without a report request
- Give the lab's customers a door: a place to submit work and follow it without a phone call

The template point carries more weight than it looks. A template written in one sitting is a guess; a template after ten runs is a standard. Software that makes the lab wait for a vendor to change a form is software whose templates stay guesses, because the change never gets requested. Your forms should be yours: changed by the people who know what needs changing, in minutes, without a release, a ticket or a developer.

The dashboard point is the one middle labs underrate. Most of what a lab wants to know about itself is already in its records and nobody looks, because looking means running something. Put the figures on the screen the team opens anyway and the lab starts managing from its own data.

How should the spreadsheet come with you?

Six years of spreadsheet is six years of the lab's history, and the middle lab cannot afford to lose it or to retype it. So the import matters more here than in any other kind of laboratory, and it is where most

migrations do their damage. A file loaded blind creates duplicates of every sample that was already entered by hand, mis-reads a date column as the wrong month, and puts a value in the field beside the one it belonged in.

An import should show its work before it writes anything. Choosing a file should only read it. The person importing should see, before pressing anything, how many records would be created, how many would update records that already exist, how many are already current, and every row the software cannot use with the row number, the column at fault and the reason in words. Where an existing record disagrees with the file, each field should be shown side by side with a choice to keep or replace. Nothing should be written until the person says so.

Two properties follow. The import should match on a column the lab controls, so that re-importing a corrected file next month updates rather than duplicates. And the mapping from column to field should be done once, by someone who knows both the spreadsheet and the system, and then run by whoever has the file, over and over, without redoing the thinking. That split is what turns migration from a one-off project into an ordinary Tuesday task, and it is the difference between migration and damage.

The data migration that a project-shaped LIMS bills as a service is, for the middle lab, a screen its own people should be able to use.

What do customers see?

The middle lab's customers telephone. They ask whether the samples arrived, when results are due, and whether last month's certificate can be sent again. Each call is answered by someone who has to stop what they were doing, find the answer in the spreadsheet and read it out. It is the most expensive way a laboratory can share information, and the spreadsheet offers no alternative.

Off-the-shelf laboratory software should give customers a door of their own. A place to submit work so it arrives already structured, with the fields the lab needs filled in by the person who knows the answers. A place to follow progress, so "has it arrived" and "when is it due" are answered by a screen. A place to collect

reports, so the certificate is fetched rather than re-sent. Customers who can see their own work stop phoning the lab, and the lab's people get their afternoons back.

What the customer should not see is the lab's internal working: other customers' samples, the analyst's notes, the review that has not yet happened. The door should show each customer their own work and nothing else, enforced where the data is read rather than by what a screen chooses to display. A customer portal that requires the lab to publish each result by hand is a second inbox, not a door.

What a platform in the middle carries

LabLynx has implemented project-shaped LIMS for large laboratories for decades, and the middle lab is the one we watched keep its spreadsheet for another year because none of our quotes fitted it. The platform described below is the answer to that lab, and the boundary is stated plainly.

> **How LabLynx handles this**
>
> LabLynx One is off-the-shelf laboratory software for the lab in the middle: records with real fields, templates the lab's own people write and change in minutes, applications switched on per working group, an audit trail kept for everything, dashboards a team arranges for itself, and a customer door. The spreadsheet import should show what it will create, update and skip, row by row with reasons, and write nothing until confirmed. It is not a validated enterprise LIMS for a multi-site pharmaceutical QC operation with forty instrument interfaces, and if that is your lab we will say so and quote you the project. If your lab has structure and no project team, this is the work we do.

Frequently asked questions

When has a lab outgrown spreadsheets?

When someone asks the spreadsheet a question it cannot answer: who changed this cell and what did it hold before, who reviewed this result and when, which samples from one customer were run on one

instrument last quarter. Those questions arrive with accreditation, a customer audit or a complaint. The size of the spreadsheet is not the signal. The first unanswerable question is.

Is there a LIMS for small laboratories?

There is a category between the spreadsheet and the LIMS project, and it fits the small and mid-sized lab better than a scaled-down enterprise system. Off-the-shelf software gives the lab records with real fields, templates its own people write, an audit trail, a spreadsheet import and a customer door, usable in days. It leaves out the six-month configuration project the small lab could not staff.

How long does off-the-shelf lab software take to set up?

Days rather than months, because the lab's own people do the fitting. Writing the first templates, importing the existing spreadsheet and arranging a dashboard is work measured in afternoons, and the templates then improve over the first ten runs. What takes months in a LIMS project is configuration, validation and migration; off the shelf means those are the lab's own tasks made small.

Can I import my existing spreadsheets?

You should be able to, and the import should show its work first. Choosing the file should only read it, and the screen should say how many records would be created, how many would update existing records, how many are already current, and which rows cannot be used and why, before anything is written. Re-importing a corrected file should update rather than duplicate.

Do I still need validation for a small lab system?

If the lab works under a regulation or an accreditation standard, yes, but validation is proportionate to risk, not to the size of the vendor's documentation. Software the lab fits itself needs the lab to document what it set up and show that the records it relies on behave as intended. That is smaller than validating a bespoke configuration, and it is the lab's task.

Sources and references

1. ISO/IEC 17025:2017 General requirements for the competence of testing and calibration laboratories, clause 7.11

   International Organization for Standardization

2. Data Integrity and Compliance With Drug CGMP: Questions and Answers

   U.S. Food and Drug Administration

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/migrating-off-spreadsheets-to-a-lims/

Downloaded September 11, 2026. LabLynx builds laboratory informatics software: LIMS, ELN, LIS and lab automation.