

How to Build LMS RFP Requirements Without Losing Your Mind

Even When It Feels Like Herding Cats



A practical guide for organizations selecting their next Learning Management System

SUMMARY

This practical guide explains how to build LMS RFP requirements without the chaos that derails so many LMS selection projects. It explores the common mistakes organizations make during requirements gathering and why stakeholder alignment is often so difficult. You'll also learn how to create a more focused, structured process that leads to better LMS decisions and fewer regrets later.

Table of Contents

01	Why This Feels So Hard (It's Not You) Everyone is right. That's the problem.	4
02	Where Requirements Go Off the Rails Requirements gathering rarely fails dramatically. It drifts.	5
03	The Mistakes That Come Back Later Looks good on paper. Reality, not so much.	6
04	A Framework That Brings Order to the Chaos A good plan doesn't control the cats. It gives them a direction.	7
05	Practical Tools That Actually Help Better tools. Better insights. Fewer surprises later.	8
06	What People Who've Been Through This Say The pattern in these observations is consistent.	9
07	Before You Open the Gate Pause. Check. Align. Then open the gate.	11
08	Bringing the Herd Into Town The goal isn't perfection. It's control.	12
09	Key Takeaways Fix the process, and you dramatically improve the outcome.	13
10	FAQs	14

INTRODUCTION

There's a moment, early in most LMS selection projects, when someone says, "Let's just gather requirements."

It sounds reasonable. Responsible, even. And then the invites go out. HR joins. Other departments join. IT joins. Compliance joins. Someone from sales shows up. Someone else brings a spreadsheet. Another person forwards an email thread that started three months ago.

By week two, the document is 14 pages long. By week four, it's 42. By week six, no one is quite sure what problem you're solving anymore.

This is the part no one warns you about. Selecting an LMS is not a technology problem. It's a coordination problem. Or, to borrow from a certain group of fictional cowboys, it's herding cats.

And not the polite, domesticated kind. The half-wild, dart-in-every-direction, scratch-you-on-the-way-out kind.

60%

According to Gartner, 60% of enterprise software buyers regret their purchase within 12–18 months. Not because the product didn't work, but because the requirements never really did.

This guide is designed to help you do something deceptively difficult: Bring the herd into town and not lose half of it along the way.

01

SECTION

Why This Feels So Hard (It's Not You)

Most enterprise software solves a specific problem. An LMS solves several.

HR sees onboarding

Compliance sees audit risk

Sales sees enablement

IT sees integrations

Executives see ROI

Learners see something they hope they don't have to log into too often

Everyone is right. That's the problem.

In most LMS decisions, you're not just defining requirements. You're bringing together a group of people who each see the system differently, each has valid concerns, and each is optimizing for something else.

What you're really managing isn't a requirements process. It's a temporary coalition of stakeholders who don't naturally agree on anything, trying to define a shared future system.

If that feels chaotic, it's because it is.

02

SECTION

Where Requirements Go Off the Rails

Requirements gathering rarely fails dramatically. It drifts. Slowly. Quietly. Until the document no longer reflects reality.

Here's how it usually happens.

1 The Wish List Problem

You ask what people need. They tell you everything they've ever wanted. Gamification. AI. Dashboards. Mobile apps. Offline access in remote regions that no one currently operates in. Individually, each request makes sense. Collectively, they describe a system that does not exist.

2 The Edge Case Trap

Someone raises a scenario that happens once a year. It's valid. It matters. And suddenly, the entire evaluation begins to orbit around it. You start designing for the exception instead of the rule.

3 The Alignment Illusion

Everyone participates. Meetings happen. Notes are taken. It *feels* aligned. Until decisions need to be made. [McKinsey](#) has found that organizations that engage stakeholders early are significantly more likely to succeed. But participation is not alignment. Alignment is when people agree on what matters most.

4 Features Masquerading as Needs

People ask for tools. What they actually need are outcomes.

- Faster onboarding
- Fewer compliance gaps
- Less administrative friction

The feature is just the means.

5 The Quiet Cost of Poor Requirements

Poor requirements are one of the most common reasons projects struggle or fail. Not because teams don't care. Because they underestimate how hard this part is.

03

SECTION

The Mistakes That Come Back Later

Common Mistakes: Looks good on paper. Reality, not so much.

Most LMS regrets don't show up during selection. They show up six months after launch. And when you trace them back, they usually point here.

Mistake 1: Starting With Demos

Demos feel like progress. They're not. They're "theater".

John Leh puts it simply: "Successful LMS buyers know better. They realize they must define their requirements first."

Watch enough demos early, and your requirements start to mirror what you've seen. Not what you actually need.

Mistake 2: Trying to Make Everyone Happy

There's a moment when the team realizes: Not everyone is going to get what they want.

This is where many projects slow down. Or stall. Or quietly expand the scope to avoid the conversation.

Mistake 3: Too Much of a Good Thing

More requirements feel safer. It's not.

The more you add, the harder it becomes to see what actually matters. RFPs get longer, responses get heavier, and the process takes longer. Clarity beats completeness every time.

Mistake 4: Forgetting the Learner

It's surprisingly easy to design an LMS that works beautifully for administrators and poorly for everyone else.

eLearning Industry data consistently shows user experience as a top driver of dissatisfaction.

Clarity beats completeness every time.

04

SECTION

A Framework That Brings Order to the Chaos

The Framework: A good plan doesn't control the cats. It gives them a direction.

When things feel chaotic, structure helps. No more discussion. Structure.

THE CLEAR MODEL



Clarify Objectives

Start with outcomes. If you can't explain what success looks like, requirements won't fix that.



Limit Stakeholders

Not everyone needs to be in every conversation. A smaller, focused group moves faster and decides more clearly.



Establish Use Cases

Real workflows beat hypothetical scenarios. If it doesn't happen often, it shouldn't drive the decision.



Assign Ownership

Every requirement needs someone accountable. Otherwise, it belongs to everyone, which means no one.



Rank Priorities

Must Have. Should Have. Nice to Have. This is where the real work happens. And where most teams hesitate.

Must Have

Should Have

Nice to Have

05

SECTION

Practical Tools That Actually Help

Tools & Techniques: Better tools. Better insights. Fewer surprises later.

This is where things get easier. Not effortless. But manageable.

1 Interviews Over Surveys

Surveys collect answers. Interviews uncover thinking. Ask: "What problem are you trying to solve?" Then keep asking why.

2 Scenario Workshops

Instead of listing features, walk through real situations. "How does onboarding actually happen today?" This is where the useful detail shows up.

3 The Decision Log

Write things down. Not just the requirement. The decision behind it. The future you will be grateful.

4 The Parking Lot

Not every idea belongs in this phase. But every idea should be acknowledged. This keeps conversations moving.

5 Pilot Testing

Demos are controlled environments. Real users are not. A pilot reveals what demos don't.

Ask: "What problem are you trying to solve?" Then keep asking why.

06

SECTION

What People Who've Been Through This Say



I've seen organizations list 80 must-have features, which just tells vendors you haven't done your homework. Keep your must-haves to 10 to 15 core requirements.

ProProfs Training Blog



The three major reasons a project will succeed are user involvement, executive management support, and a clear statement of requirements. Without them, the chances of success are greatly diminished.

The Standish Group, CHAOS Report



Successful LMS buyers know better. They realize they must define their requirements first. Talking with software providers before clarifying requirements creates confusion, wastes time, and triggers a flood of follow-up activity.

John Leh, CEO and Lead Analyst, Talented Learning



Projects with clear requirements documented before development started were 97% more likely to succeed.

Dr. Junade Ali, author of Impact Engineering, citing research from a study of 600 software engineers (The Register, June 2024)



With more than 1,000 learning systems available today, it's easy to become a victim of Parkinson's Law, where work expands to fill the time available for its completion.

John Leh, CEO & Lead Analyst, Talented Learning (2025)



Avoid vague language like 'user-friendly' or 'intuitive interface.' Instead, use concrete, testable criteria. For example: 'The system must support 5,000 concurrent users without page load times exceeding 3 seconds'.

ProProfs Training Blog



The danger of selecting the incorrect solution is decreased by working with the appropriate stakeholders from the beginning and continuing to involve them throughout the implementation project.

Olive Technologies



Most LMS failures come from internal misalignment, not vendor limitations. The RFP should reflect shared priorities, not one department's wishlist.

ProProfs Training Blog

The pattern in these observations is consistent: organizations that invest in clear, structured requirements make better decisions. Those who rush, improvise, or skip structure almost always end up in a replacement cycle.

3.5 / 5

The **Brandon Hall Group** research adds a sobering note: satisfaction scores across 17 LMS factors rarely exceeded 3.5 out of 5. The chronic dissatisfaction isn't purely a product problem. It's a mismatch problem, and the mismatch starts with requirements.

07

SECTION

Before You Open the Gate

Before you issue an RFP. Pause. Check. Align. Then open the gate.

Before issuing an RFP, pause and ask:

☐ Do we understand our core workflows?

☐ Do we know what's non-negotiable?

☐ Are we aligned on priorities?

☐ Do we know what success looks like?

If the answer is “not quite,” don't open the gate yet.

08

SECTION

Bringing the Herd Into Town

The goal isn't perfection. It's control.

If you:

Focus on outcomes

Prioritize clearly

Align stakeholders

Test real scenarios

You can do something surprisingly rare. You can bring the herd into town and not lose a single one.

And if you've ever tried to herd cats, you know that's no small thing.

LMS RFP REQUIREMENTS TEMPLATE

Get the RFP Template by Craig Weiss

Compiled by an independent LMS analyst, this comprehensive RFP template is organized into nine easy-to-use Excel tabs, allowing you to set your criteria before the first demo, send the same questions to every vendor, and see exactly how they stack up.

Features and functionality

Multi-tenancy and portals

Skills

Cohort-based learning

Mentoring

Technology and security

Training and support

Vendor and system information

Your information

For more information, visit expertusone.com/lms-rfp-requirements-template

Takeaways That Will Save You From Regret

1 Requirements gathering is where most LMS decisions fail. Not vendor selection.

The [Gartner research](#) is detailed: 60% of software buyers experience regret within 18 months, and the causes almost always trace back to planning failures, not product failures. Fix the process, and you dramatically improve the outcome.

2 Structure your stakeholder engagement.

Don't invite everyone to one meeting and call it requirements gathering. Run targeted sessions by functional group, use structured templates, set hard deadlines, and validate with real users. Inclusive does not have to mean inefficient.

3 Use cases, not feature lists.

Your vendor evaluation should be built around specific scenarios your organization actually needs to execute. If a vendor can't demo your use cases convincingly, they won't perform them reliably after implementation either.

4 Don't rush the timeline.

Allocate 6–12 months. Rushed decisions produce regret. The cost of taking three extra months to get requirements right is a fraction of the cost of a failed implementation or a premature replacement cycle.

5 Plan for three to five years, not just today.

The LMS you choose should be able to grow with your organization. Requirements that only reflect current needs are requirements that will break within 24 months.

FAQs

Q1 How long should the LMS requirements gathering process take?

For most organizations, 4–8 weeks for requirements gathering is realistic, within a total selection-to-go-live timeline of 6–12 months. Rushing this phase to save time almost always costs more time later through rework, re-evaluation, or premature replacement.

Q2 Who should be involved in gathering LMS requirements?

At minimum: L&D/Training, HR, IT, and one operational leader with budget authority. Depending on your use cases, you may also need input from Compliance/Legal, Sales Enablement, Customer Education, and representative learners from key populations. Keep your core team small (5–8 people). Run structured input sessions with a broader group. Don't try to include everyone in every decision.

Q3 How do we handle stakeholders who want to add requirements after the deadline?

Log their input in the Parking Lot document and communicate that it will be considered for a future phase. Be consistent about the cutoff. If you allow late additions, you signal that the deadline doesn't matter and the process never ends.

Q4 Should we use an RFP or just do demos?

Both, in the right order. Requirements first. Then a focused RFP is sent to a shortlist of 3–5 qualified vendors. Then vendor demos based on your specific use cases. Skipping the RFP means relying on vendor-controlled demos and verbal promises, without written accountability.

Q5 How many vendors should we evaluate?

Start broad (research 10–15 options using your requirements as filters), create a shortlist of 3–5 for the formal RFP process, and plan for 2–3 to move to the demo and pilot stage. Evaluating more than 5 in-depth leads to decision fatigue and shallower assessments.

Q6 What's the difference between a requirement and a preference?

A requirement is something your organization cannot function without, or that creates a legal/compliance risk if absent. A preference is something you'd like to have that would make the system better. The [MoSCoW framework](#) is the best tool for making this distinction explicit and defensible.

Q7 How do we prevent scope creep during the requirements phase?

Use a structured input process with a defined template, set and enforce a hard deadline for submissions, and create a visible Parking Lot for out-of-scope items. Communicate clearly that the deadline exists to protect the team, not to exclude anyone's input.

Q8 What if we don't currently have an LMS?

If this is your first LMS purchase, your current-state audit looks different: document all existing training processes (spreadsheets, in-person sessions, third-party platforms, email-based delivery) and identify where they break down. Your use cases will focus on what your team needs to start doing, not just what they need to improve.

Q9 How do we validate that our requirements are realistic?

Run a lightweight market scan before finalizing requirements. Do most LMS platforms support your Must Haves? If something appears on very few platforms, it's either a niche requirement (worth keeping if critical) or a hypothetical edge case (worth moving to the Parking Lot). You can also have early informal conversations with 2–3 vendors specifically to pressure-test your requirements. Not to demo, but to sanity-check feasibility.

Q10 What happens if we discover we chose the wrong LMS after go-live?

First, distinguish between a requirements failure (the system can't do what you need) and an adoption failure (the system can do it, but people aren't using it). Adoption failures are often fixable through training and change management. Requirements failures are much harder. If you're in a genuine requirements failure, document specifically what's missing, assess whether it can be addressed through configuration, integration, or vendor development, and set a realistic timeline for either a remediation plan or a replacement cycle, this time with a structured requirements process.



info@expertusone.com
www.expertusone.com
877-827-8160