
Build vs. Buy: A Volunteer Leader's Guide to Getting Time Back

Four questions to answer before you decide how to fix volunteer admin.

BUILD VS. BUY

The Week Nobody Budgeted For

It's the week before your biggest volunteer intake of the year. Two thousand applications have come in against a screening workflow sized for a few hundred. Forty of this week's shifts span three different states, each with its own compliance requirement, and there's no single view of who's actually cleared to work which one. The report the board asked for last week means someone manually reconciling exports from two systems that were never built to share a single source of truth.

By Monday, you're only a third of the way through applications and now your volunteer manager is off sick.

This is not a story about an under-resourced team. It's what happens when a volunteer programme scales past the system, or patchwork of systems, that was built to run it. And it's the backdrop to four questions that come up in almost every conversation we have with volunteer leaders, operations heads, and the finance and executive teams who eventually sign off on fixing it:

- 01 How much time is admin actually costing your coordinators each week?
- 02 What does best-practice onboarding look like for a programme your size?
- 03 Have you actually outgrown the system you're on?
- 04 Should you build your own system, or buy one?

This guide takes you through each of these questions with evidence, not just our own two cents!



1. The Hidden Cost of a Coordinator's Week

Ask almost any volunteer coordinator about their biggest challenges and it's rarely the volunteers themselves. It's the admin that gets in the way of supporting them: chasing compliance paperwork, confirming shifts one by one, or pulling together an impact report at short notice.

Sector research puts a number on what most coordinators already feel. Nonprofit workforce reporting has estimated that organisations managing 200 or more active volunteers see **coordinators spend somewhere in the region of 15–20 hours a week on volunteer administration** (scheduling, communication and tracking) rather than on the volunteers themselves (Nonprofit Times Workforce Report, 2025). Separate sector surveys have found **administrative tasks consuming well over half of a typical coordinator's time**. That's roughly half a role, per coordinator, spent on work that doesn't require a human judgement call.

FOR FINANCE

Put a figure next to it. A coordinator on a \$70,000 loaded salary losing even the lower end of that range to manual admin represents somewhere around \$12,000–\$16,000 a year spent on tasks a platform can automate, before counting the volunteer attrition that slow, manual admin tends to cause. That's the number worth sitting next to any software cost comparison, not just the licence price.

Worth asking your own team:

- How many hours a week do coordinators actually spend on shift confirmations, compliance chasing, and manual reporting?
- If that admin disappeared tomorrow, what would coordinators do instead?
- Has anyone measured this, or is it still an estimate nobody's tested?

2. Best Practice Looks Different Depending on How You're Structured

At the scale we're talking about (500 or more volunteers across programs, sites, or entities) "best practice onboarding" isn't one workflow. It depends on the shape of your organisation, not just its size.

Single entity, high-volume programme

One organisation, one governance structure, but a volunteer base large enough that manual onboarding creates a permanent backlog. **Best practice here is a single automated sequence. Application, screening, document upload, training, first-shift confirmation, that scales without adding headcount every time volunteer numbers grow.**

Multi-entity or federated structure

With national bodies, state or regional associations, and local chapters or clubs, compliance requirements can vary at every level. **Best practice is a tiered approach: local entities manage their own onboarding while adhering to national standards, with centralised visibility for reporting and oversight, without manually reconciling every regional variation.**



Peak-burst, event-driven deployment

Thousands of volunteers may need to be mobilised within a short, fixed window, where onboarding failures become a safety and reputational risk, not simply an inconvenience. **Expo 2023 Doha processed 50,000 volunteer applications in just 72 hours, demonstrating the importance of systems that can perform under concentrated demand.**

The same challenge exists at smaller scales. For FareShare I Second Bite a merger saw manual onboarding stretch to 4–10 weeks. Automation significantly reduced that cycle and, importantly, removed the need for coordinators to chase every applicant manually.

Best practice isn't just a process that works. It's one that continues to work when demand spikes.

FOR EXECUTIVES

What matters at board level isn't the onboarding workflow itself. It's what a broken one exposes you to: an unscreened volunteer working with vulnerable people, a compliance document nobody actually checked, an incident that becomes a governance conversation rather than an admin footnote. Ask your team plainly: could an unverified volunteer currently end up on a shift, and how would we know before it became a problem?



3. Have You Outgrown Your Current System?

By the time build vs. buy becomes a live conversation, most organisations aren't starting from nothing. There's already a system in place: a homegrown database, a generic scheduling tool stretched well past its original brief, or an early volunteer platform that fit the organisation at a much smaller size.

The real question isn't whether you have a system. It's whether the one you have was built for the volume, site count, and compliance load you're managing today.

A few signals tend to show up before anyone puts it in those terms:

- 1 Reporting that used to take an hour can stretch into a week as your current system isn't built to turn complex volunteer data into meaningful insights.
- 2 A growing list of manual workarounds that only one or two people fully understand.
- 3 New sites, programmes, or partner organisations that don't fit the system's original structure, so they get bolted on rather than properly supported.
- 4 Using multiple systems to create a workaround with no ability to integrate with your CRM, HRIS, or finance system.
- 5 Onboarding is largely manual and becoming unsustainable. With different roles and locations requiring their own screening and compliance workflows, your team can no longer manage the complexity efficiently.

None of these break the system outright. They just make it progressively more expensive to run, in people's time if not in the budget line.

Sector data backs this up at scale: adoption of dedicated, purpose-built volunteer management platforms has grown from roughly a quarter to closer to two-fifths of mid-size nonprofits over a recent two-year period (NTEN Nonprofit Technology Benchmarks).

It's a shift driven less by any one system failing outright, and more by organisations reaching the point where what they have can't absorb the next stage of growth.

Three questions cut through the noise faster than a feature audit:

- Are you adding headcount to manage the system, not the mission? Staff whose real job has become working around the tool's limits?
- Is every new integration or region a bespoke project rather than a configuration change?
- Could anyone tell you with confidence what would break if your busiest week doubled overnight?

FOR OPERATIONS & FINANCE

"It still basically works" isn't the same test as "it still fits." A system that was right-sized at 150 volunteers on one site can quietly become the actual bottleneck at 600 across five. Not because it broke, but because every workaround built to stretch it now has to be maintained by hand. That maintenance cost rarely shows up on a budget line, which is exactly why it's worth putting a number on before it becomes someone's full-time job.



4. Build or Buy: The Real Trade-off

Once the time cost and the continuity risk are on the table, the harder question follows: build something custom, or buy an existing platform?

PROS OF BUILDING	CONS OF BUILDING
Full control Tailor the system exactly to your workflows and processes, with no vendor constraints.	Requires real capacity Software development isn't a side project. You need someone to lead build, QA, integration, and training.
No recurring licence fees You own the outcome outright once it's built.	You're starting from zero Purpose-built platforms carry years of cross-organisation learning. Matching that from scratch, fast, is hard.
Timing is entirely yours You decide scope, sequencing, and spend.	Upkeep never stops Security patching, bug fixes, and feature updates are a permanent line item, not a one-off cost.
No vendor dependency Updates and fixes happen on your schedule, not someone else's roadmap.	Unproven until it isn't You'll be trialling your own system for the first time with your team absorbing the cost of any early mistakes.

What building actually costs beyond the build

Development is the visible cost. The recurring costs that catch teams out are the ones nobody budgets for at the outset: ongoing security patching, technical support, and feature updates as needs change. And if scope creeps during the build, which it very often does, a project that started as a few months' work can quietly become a multi-year commitment.

Four questions worth asking before you commit either way

- Does this genuinely need to be custom-built, or would configuring an existing platform get you most of the way there?
- Can your team realistically own security, compliance accreditation, and technical support for the life of the system, not just the build?
- What's your actual timeline pressure? If you need something running this year, a multi-year build is the wrong starting point regardless of its long-term appeal.
- Have you priced the full lifecycle cost of building including the team needed to keep it running against the full lifecycle cost of buying, not just each option's sticker price?

For most organisations without an existing, well-resourced software development function, buying carries meaningfully lower risk. Building can still be the right call where requirements are genuinely unusual and the in-house expertise exists to sustain it long-term but that's a smaller group of organisations than the case for building usually assumes.



Before You Talk to a Single Vendor

Whichever way you're leaning, these four steps apply before you go anywhere near a vendor comparison:

- 01** Audit actual weekly admin hours per coordinator. Not an estimate, a real measurement over 2–3 weeks.
- 02** Map where a delay in that admin costs you a volunteer, not just a coordinator's time.
- 03** Check whether your current tooling was built for today's volunteer volume, or a much smaller version of your program.
- 04** Get Finance and Operations aligned on total lifecycle cost (build or buy) before anyone starts evaluating specific options.

None of this tells you which specific platform is right, or whether buying is definitely the answer for your organisation. That depends on your size, your requirements, and your appetite for owning long-term technical risk. What it should do is give you a straight answer to the four questions that come up in almost every one of these conversations, before you're deep in vendor demos and comparison spreadsheets of your own.

If it's useful to keep tracking how volunteer-sector organisations are working through this decision, Rosterfy publishes ongoing research and case studies on volunteer operations at scale.



ABOUT ROSTERFY

Rosterfy is a leading volunteer management platform trusted by nonprofits, hospitals, sports organisations and major events across 35 countries. From hundreds to tens of thousands of volunteers, Rosterfy helps organisations manage, engage and retain their volunteer communities at scale.

[LEARN MORE](#)