

8-Figure Systems

Software Strategy for Replacing
Costly, Off-the-Shelf Tools

For businesses doing \$10M+ in
revenue who are running critical work
with enterprise tools.





1

The Problem

You Didn't Choose
This Software Stack.



2

4 Choices

Everything Comes Down
to These 4 Choice



3

Quiz & Results

Which Systems Strategy
Matches Your Business?



SECTION 1: THE PROBLEM

You Didn't Choose This Software Stack. It Accumulated Over Time.

No one sat down and designed this setup. It built up one workaround at a time, while you were focused on running the business. Now, keeping things moving means working around your tools instead of with them, and those workarounds have quietly become part of employees' jobs.



Should You Follow Their Leads?

AI has made custom software cheaper to build for businesses of every size, which means renting a patchwork of tools is a worse deal today than it was two years ago.

Starbucks Slashes IBM Expense

Starbucks is building custom software to replace parts of its Microsoft and IBM stack, with AI-assisted development playing a key role, according to an internal presentation reported by Bloomberg. The company spends roughly \$400M a year on software and is targeting \$2B in cost cuts. Its CTO, Anand Varadarajan, told employees there are “clear opportunities to reduce the spend in software.”

- Point of sale system (POS)
- Inventory management system
- Maintenance management system

Klarna Cut 1,200+ SaaS Tools

Klarna eliminated more than 1,200 third-party SaaS expenses in favor of an internally built, AI-powered tech stack. CEO Sebastian Siemiatkowski has framed it as a straightforward build-versus-buy decision. AI made it practical to consolidate scattered SaaS data into one internal system instead of paying for dozens of disconnected tools to each handle a piece of the job.

- Salesforce CRM
- Workday HR
- 1,200+ other platforms



Signs It's Time to Axe Your SaaS

01

A spreadsheet is running a critical workflow, and only one person understands it.

If that person is out sick, on vacation, or leaves the company, the workflow stops with them.

02

The same data is typed into multiple different tools by hand weekly.

Every manual entry is a chance for a typo to become a real cost.

03

Your tools don't talk to each other, so someone has to be the human API.

That person's job has quietly become moving information between systems, not the job they were hired for.

04

You've built manual workarounds nobody has time to remove.

Each one made sense in the moment. Together, they're a second, invisible system running the business.

05

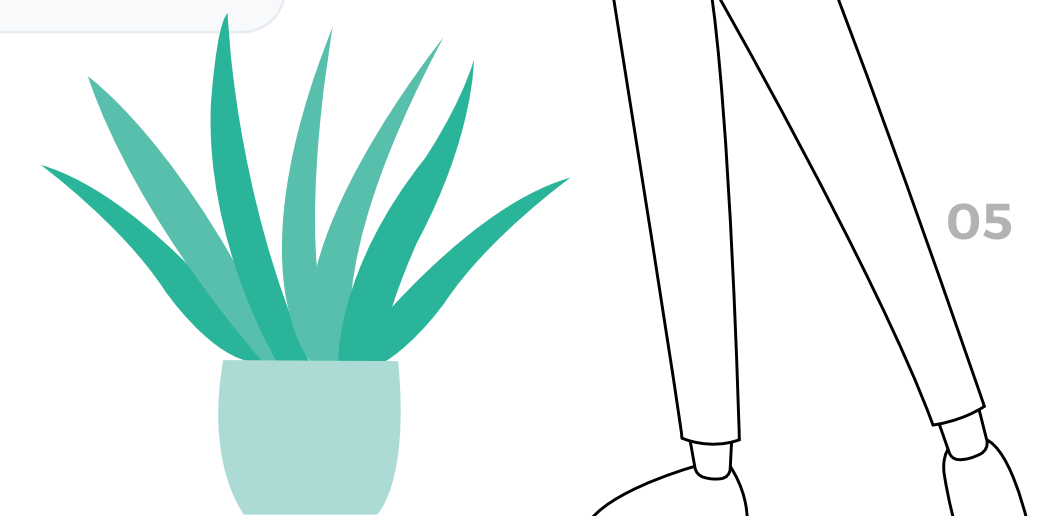
Adding a new person to the team means re-explaining the workaround, not the job.

Onboarding takes longer, and new hires learn tribal knowledge before they learn the actual role.

06

You're not sure what all your software actually costs you every month.

Between subscriptions, add-ons, and the staff time spent making it all work, the real number is usually higher than anyone expects.



The Annual Leak

Standing still always has a price tag.

Two costs stack on top of each other when a business runs on spreadsheets and stitched-together tools: what you're paying in subscriptions, and what you're losing to manual work. Most owners have a rough sense of the first number. Almost nobody has calculated the second.

What you pay in subscription costs



\$450K–\$550K/yr

(typical range for a mid-market business running 6-8 SaaS tools)

This is what you're paying just to keep the current stack running: your core platform, plus the CRM, expense tool, e-signature tool, file storage, and accounting software layered on top. Enterprise tools are often the biggest single line item, and the bill rarely goes down. Subscriptions re-price upward at renewal, and the total grows automatically as the business grows, whether or not usage does.

What you pay in inefficiency costs



\$340K–\$420K/yr

(typical range for a mid-market business with a team split across spreadsheets and disconnected tools)

This is the cost of the work happening around the software rather than inside it: staff re-keying the same data into multiple systems, manually reconciling numbers at month-end, and chasing down information that lives in someone's personal spreadsheet. It's rarely on a budget line, which is exactly why it's easy to underestimate. A team of 15-50 people spending even a few hours a week on this kind of re-entry adds up fast.

**Range:
\$820K–\$970K
total**

These are ballpark ranges based on patterns across similar-sized businesses, not a quote for your specific company. The real number depends on your tool stack, team size, and how much of your workflow still runs manually. The exact math changes business to business, but the direction doesn't: every year this goes unaddressed, the number gets a little bigger, not smaller.

This leads to...

01 Lost competitive advantage

Competitors running on integrated systems make decisions in days that take your team weeks. Data that lives in one place gets acted on. Data that lives in six spreadsheets gets argued about.

02 Inability to scale

Every new hire, market, or product line adds another layer of manual processes rather than capacity. Growth starts to feel like more work, not more leverage, because the systems weren't built to carry the extra weight.

03 Wasted subscription costs

You're paying for tools that each cover part of the job, and paying people to cover the rest by hand. The subscriptions don't go away piecemeal when you stop needing them. They just keep renewing.

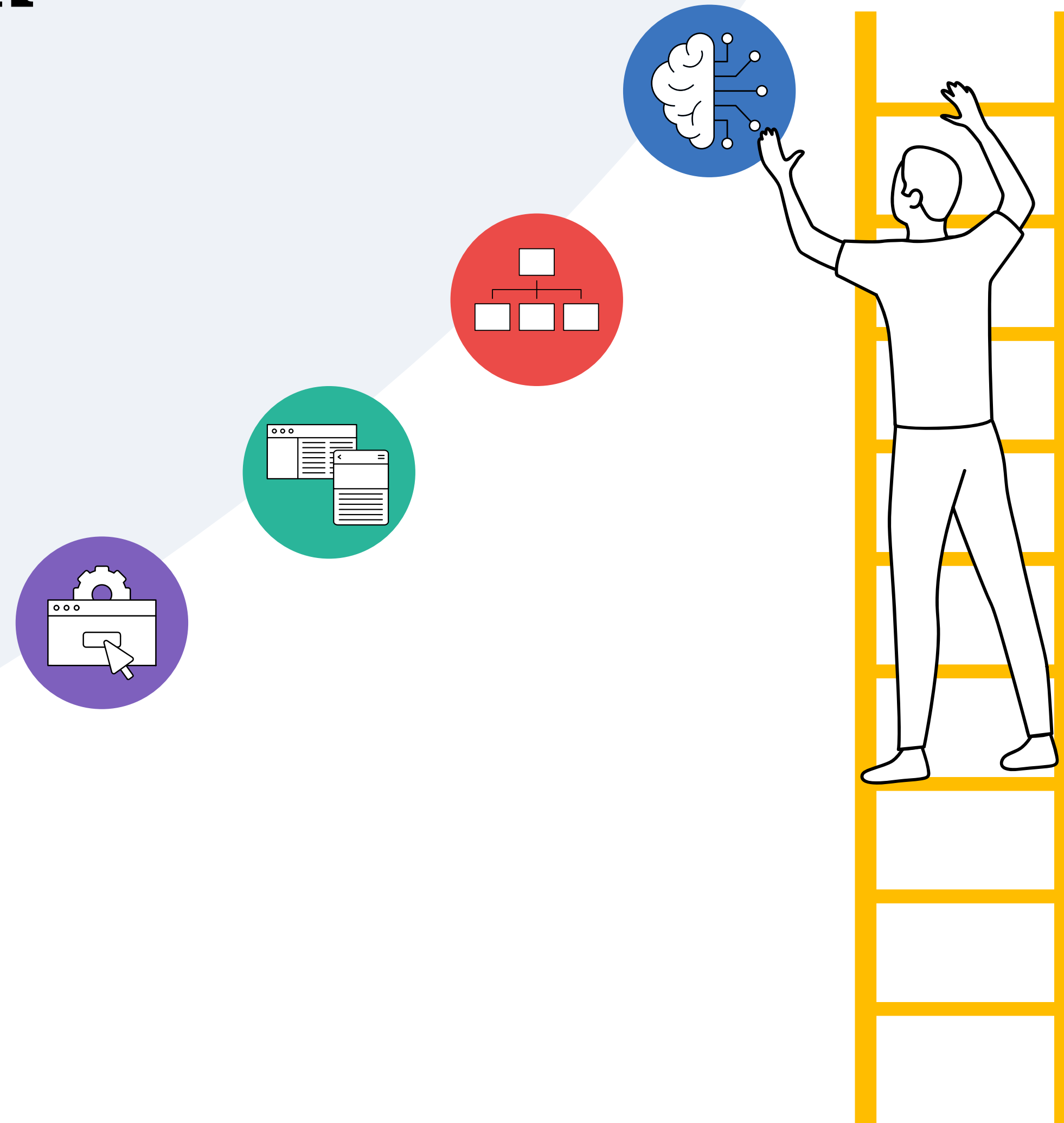


SECTION 2: 4 CHOICES

Everything Comes Down to These 4 Choices

There are four ways to manage your systems. Here's how to get the most out of whichever one is right for you.

None of these options is automatically wrong, and none of them is a permanent fix. They're all legitimate choices, but they're only right at different sizes, complexity levels, and stages of growth.



OPTION 1

Build Custom Internal Software

Best for:

Businesses at \$10M+ in revenue with workflows too specific or complex for any off-the-shelf tool

Watch out for:

The most common mistake is choosing the cheapest option instead of the right partner for the job

OPTION 2

Keep Using Spreadsheets

Best for:

Businesses under roughly \$1M in revenue, or workflows still simple enough that custom software genuinely doesn't pencil out yet

Watch out for:

It works well until the moment it doesn't, and it tends to break all at once rather than gradually

OPTION 3

Stitch SaaS Tools Together with Zapier

Best for:

Growing businesses testing out a workflow before committing to a permanent system

Watch out for:

The first few integrations feel effortless. Cost and fragility creep in fast once you're past the ninth or tenth one

OPTION 4

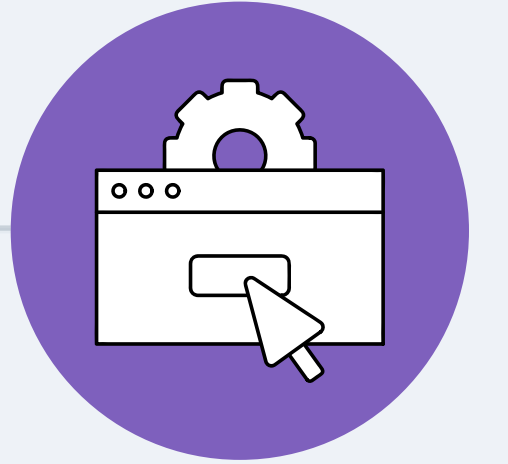
Use a No-Code / AI Tool

Best for:

Low-revenue, low-complexity situations, often a single person's side project or full-time focus

Watch out for:

Maintenance, accuracy issues, and security risk all increase once the tool starts running real business operations



OPTION 1:

Build Custom Internal Software

Your workflows are too specific for anything off the shelf.

Once a business hits real complexity, generally \$10M or more in revenue, hiring an agency to build custom software usually makes sense. It becomes the most affordable option.

Don't rush to create a finished spec or list of requirements. You need the ability to describe, in plain language, what's actually broken about how the business runs today.

Three things separate a good engagement from a bad one:

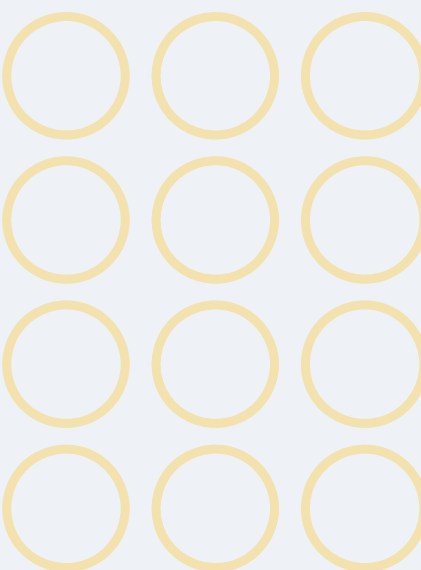
1. A partner who understands your business, not just the technical build
2. Realistic expectations about timeline
3. Treating what you build as an asset you'll maintain, not a purchase you'll forget about

A good partner helps you gain clarity. They won't expect you to arrive with it already figured out.

Be ready to describe the problem before you ask for a finished spec.

Evaluate for business understanding first, price second.

Expect payback in 1 to 3 years, with ongoing maintenance after that, not a one-time cost.



In-House VS Outsourced Development for Custom Software Projects

	In-House	Outsourced
Launch Experience	Little to none	Experienced in launching new products
Cost	High, due to the need to employ all talent	Medium to low, due to access to fractional talent
Speed	Slow, with teams working together for the first time	Medium to fast, with optimized processes
Predictability	Constant surprises and spikes in cost	Knowledgeable risk management with fixed costs

The Partner Fit Test

A partner who fails the first two questions will build what you asked for.

A partner who passes is more likely to build what you actually need.

- Do they push back on your ideas, or just take orders?
- Can they explain your business problem back to you in your own terms, not just technical ones?
- Are they the cheapest option in the room? (If yes, that's a flag, not a win.)
- Have they asked what happens after launch, not just what gets built?

The Real Timeline

1 to 3 years: The typical payback period on a custom software investment
Ongoing: the maintenance commitment after launch, not a one-time cost.

“The biggest hurdle is the patience to get that first start going. Once you push past that first barrier, everything gets smoother, and the features start delivering real value.”

— Rafael Lunardelli, CTO, DevSquad



OPTION 2:

Keep Using Spreadsheets

When spreadsheets are still the right call.

Spreadsheets get a bad reputation, but we don't judge. They're the right tool for early-stage businesses.

If your revenue is under roughly \$1M and your workflows are still simple, custom software is the wrong spend, full stop, no matter how outdated your spreadsheet feels. The real question is ROI: will fixing this save meaningful time, reduce real costs, generate revenue, or add value that shows up on the bottom line?

If you can't point to a clear yes, stick with your spreadsheets and revisit later.

A better spreadsheet is a legitimate upgrade at this stage, not a downgrade

If you can't name a specific time, cost, or revenue benefit, it's not time for custom software yet

Spreadsheet failure tends to be sudden, not gradual, and hits when you can least afford it



Spot the Tipping Point

Two signals matter more than revenue or team size:

1. Recording → Deciding. A spreadsheet that logs numbers is low risk. A spreadsheet your team prices jobs, schedules work, or reports to leadership from, especially one only one person really understands, is a different level of risk entirely.

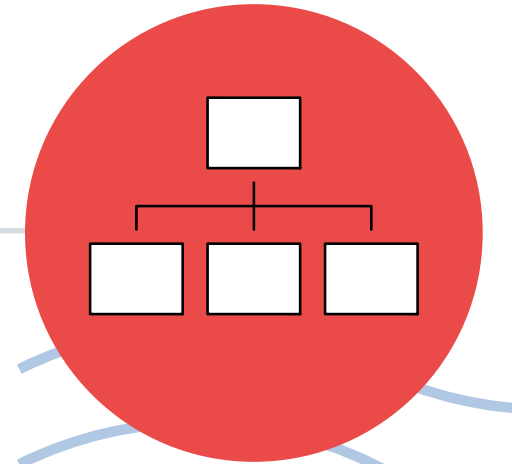
2. The shadow backup. If someone's quietly keeping a second, unofficial version to double-check the first one, trust that instinct. It usually means the tool built to save time now needs its own workaround to stay accurate.

Make It Bulletproof

While you're still in this stage, treat your most important sheet like infrastructure, not scratch paper:

1. Document how it actually works, so the knowledge isn't trapped in one person's head
2. Lock down which cells can be edited, and by whom
3. Build in a simple check to catch errors before they become decisions

None of this requires custom software. It just requires treating the sheet like the business-critical tool it's already become.



OPTION 3:

Stitch SaaS Tools Together with Zapier

Zapier makes it easy to integrate different systems.

Stitching SaaS tools together with automation helps you move fast without a dev team or a big commitment. It's a legitimate way to test out a workflow before you're sure it's the right one to build permanently.

The tradeoff shows up later than you'd expect: what starts as a handful of simple, reliable connections gradually turns into a web of dependencies that's expensive to maintain and easy to break. Every connection you build carries real complexity beneath it: security, error handling, and ongoing maintenance that a SaaS platform would normally handle for you. When you stitch tools together yourself, you inherit that complexity instead.

Integrations break on their own schedule, usually without warning. Budget real time to monitor and fix them.

The first few connections feel like magic. When that feeling fades, the setup has usually outgrown what it was built to handle.

Revisit your true costs regularly. They grow faster than any single new integration suggests.

Do the Math

Most businesses underestimate this number because it's spread across several small charges rather than a single obvious line item. Before adding another automation, total up:

Subscription cost of every connected tool + hours per month spent testing, fixing, and rebuilding broken triggers + cost of workarounds when an integration silently fails

That total, not the price of the next integration alone, is what you're actually deciding to take on.

The Breaking Point

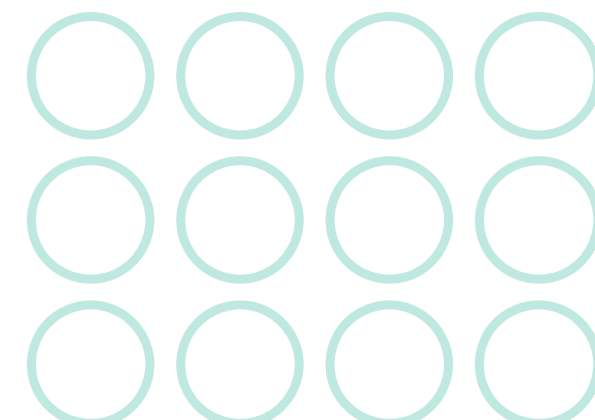
"One of the most common challenges is underestimating the complexity SaaS platforms handle behind the scenes: security, scalability, integrations, maintenance, ongoing updates."

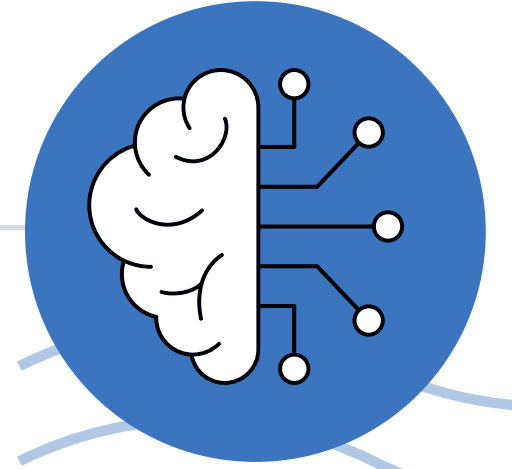
— Renato Motikawa, Technical Product Manager, DevSquad

"The first Zapier is awesome. But then you get to number ten, and now you're reworking your flows and doing hacks just to get everything to work."

— Phil Alves, CEO, DevSquad

The clearest sign you've hit that point is that more time goes into maintaining the connections than into the work they were supposed to save. One failed integration cascading into three other broken processes is a strong signal it's time for a real system, not another patch.





OPTION 4:

Use a No-Code / AI Tool

Fast to build, expensive to maintain.

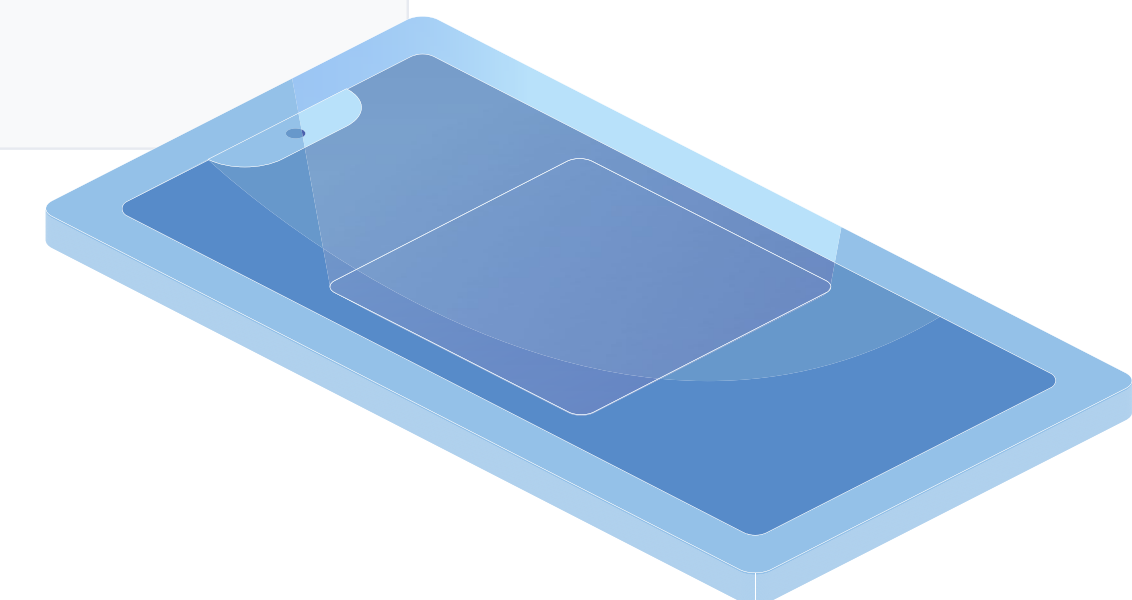
No-code and AI-assisted tools let one person stand up something real, fast, with no dev team and very little upfront cost. That's valuable at the right stage, such as a single person's side project, an early-stage idea being prototyped, or a business with no real software development needs yet.

It's not a substitute for custom development once a business has grown past that point, even though it can feel like one in the moment. Where it stops being useful is the exact moment the tool starts running workflows the business actually depends on daily. What was built quickly for one person's convenience isn't the same thing as software built to hold up under real operational weight, and the gap between the two tends to surface later than anyone expects.

This option fits low-revenue, low-complexity situations best, often a single person's project.

If daily operations already depend on it, you've likely outgrown it.

By the time the limits show up, rebuilding is usually faster than patching.



The Six-Month Cliff

Day 1:

Fast, cheap, feels like magic. One person builds something real with no dev team.

Month 6:

Small additions and quick fixes have pushed the tool past its original scope. Logic is hard to trace, changes in one place break something unrelated, and there's no documentation for anyone who'd need to take it over. Fixes take longer than expected, and the team resists changing a system they've already gotten used to, even a fragile one.

The maintenance problem never announces itself early. That's what makes it dangerous.



The Data Test

Would you be comfortable explaining exactly where your data goes and who can access it?

If the honest answer is no, that workflow doesn't belong in a no-code or AI-built tool, especially anything touching customer data, financial information, or business-sensitive material. The risk is highest for enterprise or regulated data, where a gap creates real exposure.

"A lot of the customers who come to us now got stuck on a no-code or AI-built tool. We end up starting over, because it was never real software development in the first place."

— Phil Alves, CEO, DevSquad

SECTION 3: QUIZ & RESULTS

Which Systems Strategy Matches Your Business?

Jot the letters down as you go.
Then, tally up your answers.



What’s your annual revenue?

A. Under \$1M	B. \$1M-\$10M	C. \$10M+
---------------	---------------	-----------

How many spreadsheets or tools does your team touch every day just to get work done?

A. One or two	B. Three to six	C. Seven or more, and nobody has counted lately
---------------	-----------------	---

How much of your process is unique to how you run your business, versus something any competitor could copy from a SaaS tool?

A. Mostly standard	B. Some unique steps	C. Deeply specific to us
--------------------	----------------------	--------------------------

How many people would be affected if your current system broke tomorrow?

A. Just me	B. A small team	C. Most of the company
------------	-----------------	------------------------

Are you ready to own and maintain software long-term, not just buy it once?

A. Not yet	B. Maybe, if it’s worth it	C. Yes, we treat this as an investment
------------	----------------------------	--

What’s your gut sense of your current setup?

A. It works fine for now	B. It’s starting to strain	C. It’s actively costing us deals or hours
--------------------------	----------------------------	--

If you fixed this today, what would you be most excited to get back?

A. Just some peace of mind	B. Hours back every week	C. Room to actually grow
----------------------------	--------------------------	--------------------------

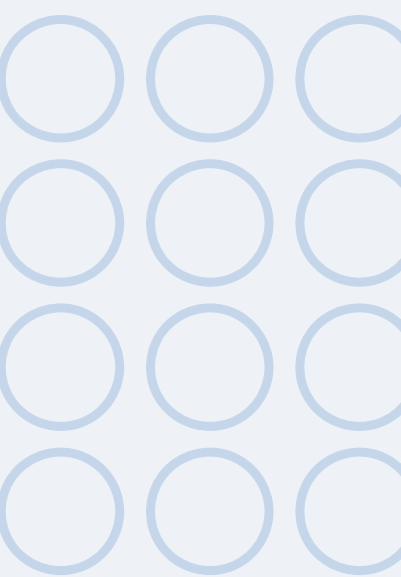


QUIZ RESULTS

Mostly A's → Spreadsheets

Right now, spreadsheets are still the right call.

Based on your answers, your business isn't at a size or level of complexity where custom software would pay for itself yet, and that's a perfectly good place to be. The move here isn't to rip out what's working. It's to make sure your current setup is documented, backed up, and not entirely dependent on one person's memory. Revisit this quiz in six to twelve months, or sooner if your revenue or team size jumps.



QUIZ RESULTS

Mostly B's → Small or Solo Team → No-Code/AI Tools

A no-code or AI tool is probably your fastest path forward.

Your answers point to a business that's outgrown a plain spreadsheet but doesn't yet need a dedicated development team. A no-code or AI-assisted tool can get you moving fast without a big upfront investment, as long as you're careful about what data you put through it. Just keep an eye on the six-month mark. That's usually when maintenance issues start to surface.



QUIZ RESULTS

Mostly B's → Larger Team → Multiple SaaS Connected with Zapier

Stitching your tools together with automation makes sense for now.

Your team is big enough that a single no-code tool probably won't hold up, but you're not yet at the size or complexity where a custom build is the clear answer. Connecting your existing tools with automation is a reasonable bridge while you figure out exactly what a permanent system needs to do. Just track your real costs as you go, and watch for the point where maintaining the connections takes more time than the work they save.

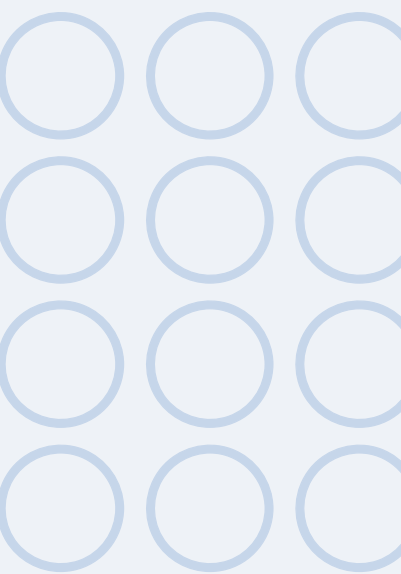


QUIZ RESULTS

Mostly C's → Build Custom Internal Software

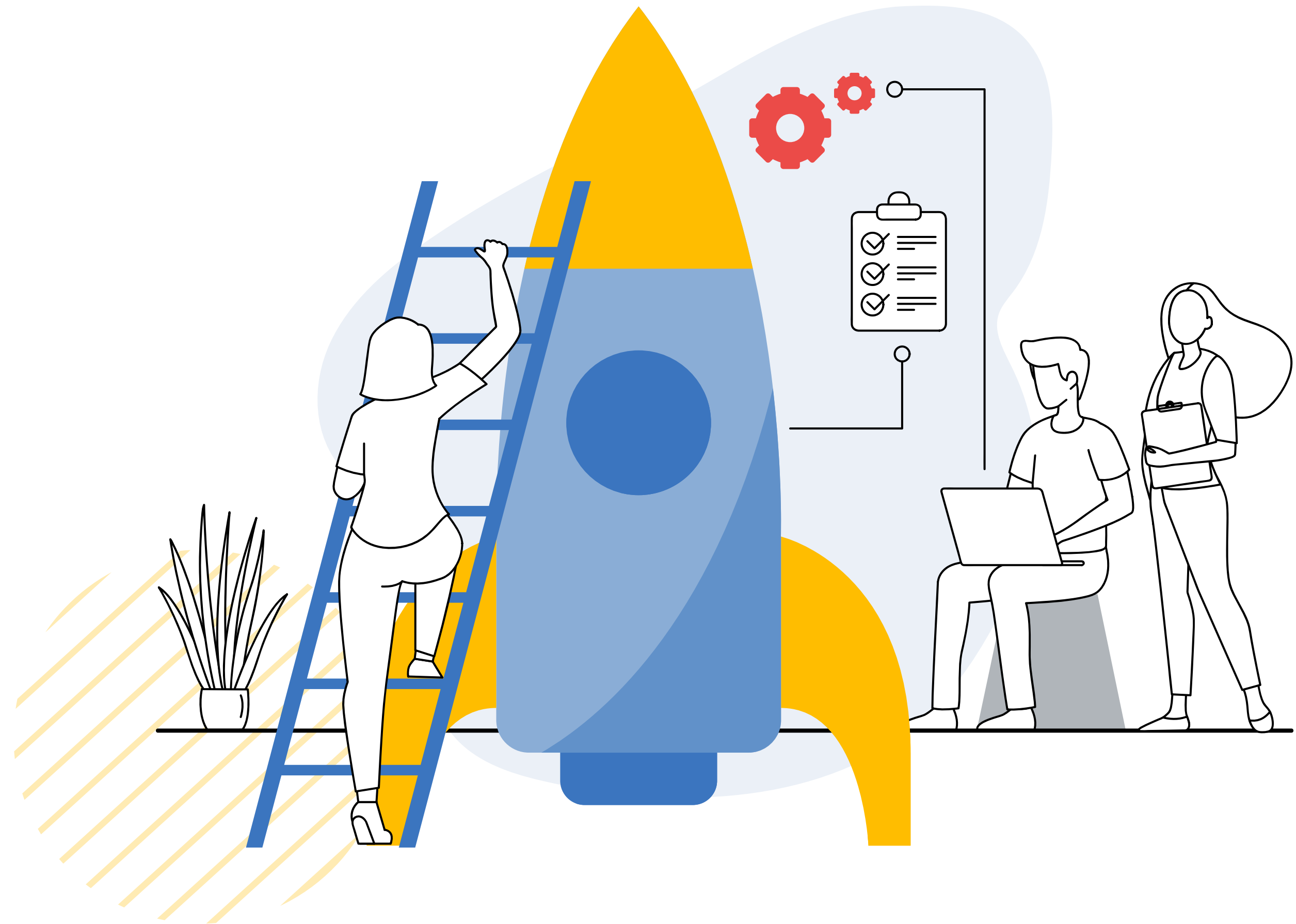
Hire an internal dev team, or better yet, a fully-managed agency.

Your business has real complexity, a team that depends on your systems working, and workflows specific enough that no off-the-shelf tool will fit cleanly. This is the point where custom software stops being a nice-to-have and becomes the option that the math actually favors. The next step should be a conversation about what's actually broken.



Why Custom Software Wins for Most 8-Figure Businesses

Here's what changes when you own the system rather than renting pieces of it.



The Anatomy of a Custom System

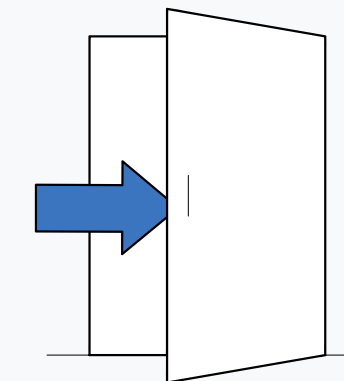


Built around you

It fits your actual workflow, instead of you fitting your workflow into someone else's product.

Valuable at exit

Owned software adds to what a buyer will pay for. A stack of subscriptions doesn't.



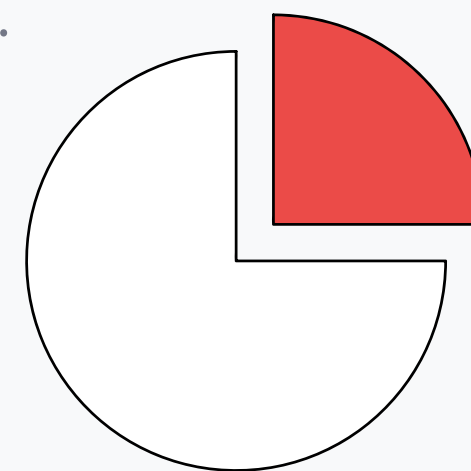
Time back, not just data back

The hours your team spends on manual re-entry become hours spent on the business instead.



No more ceiling

SaaS tools cap what you can do. Custom software removes the cap.



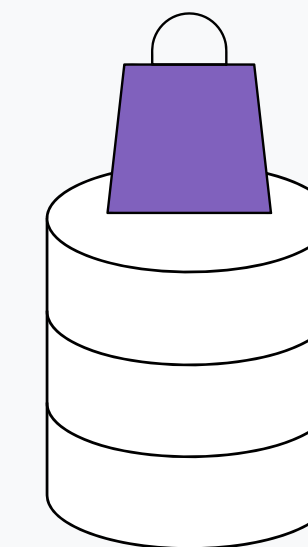
An asset, not a rental

It goes on your balance sheet. Nothing walks out the door when a contract ends.



Scales with you

It's built to carry more weight as you grow, instead of breaking under it.



Let's Compare...

Off the Shelf

Half your business runs on spreadsheets only one person understands

Subscriptions that re-price every renewal upward, forever

Tools that don't talk to each other, so someone has to be the human API

Process trapped in one person's head

Nothing to show for it at exit

Custom Software

One system your whole team can see and trust

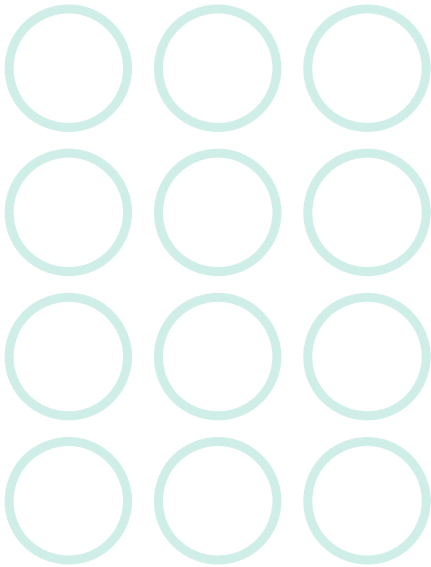
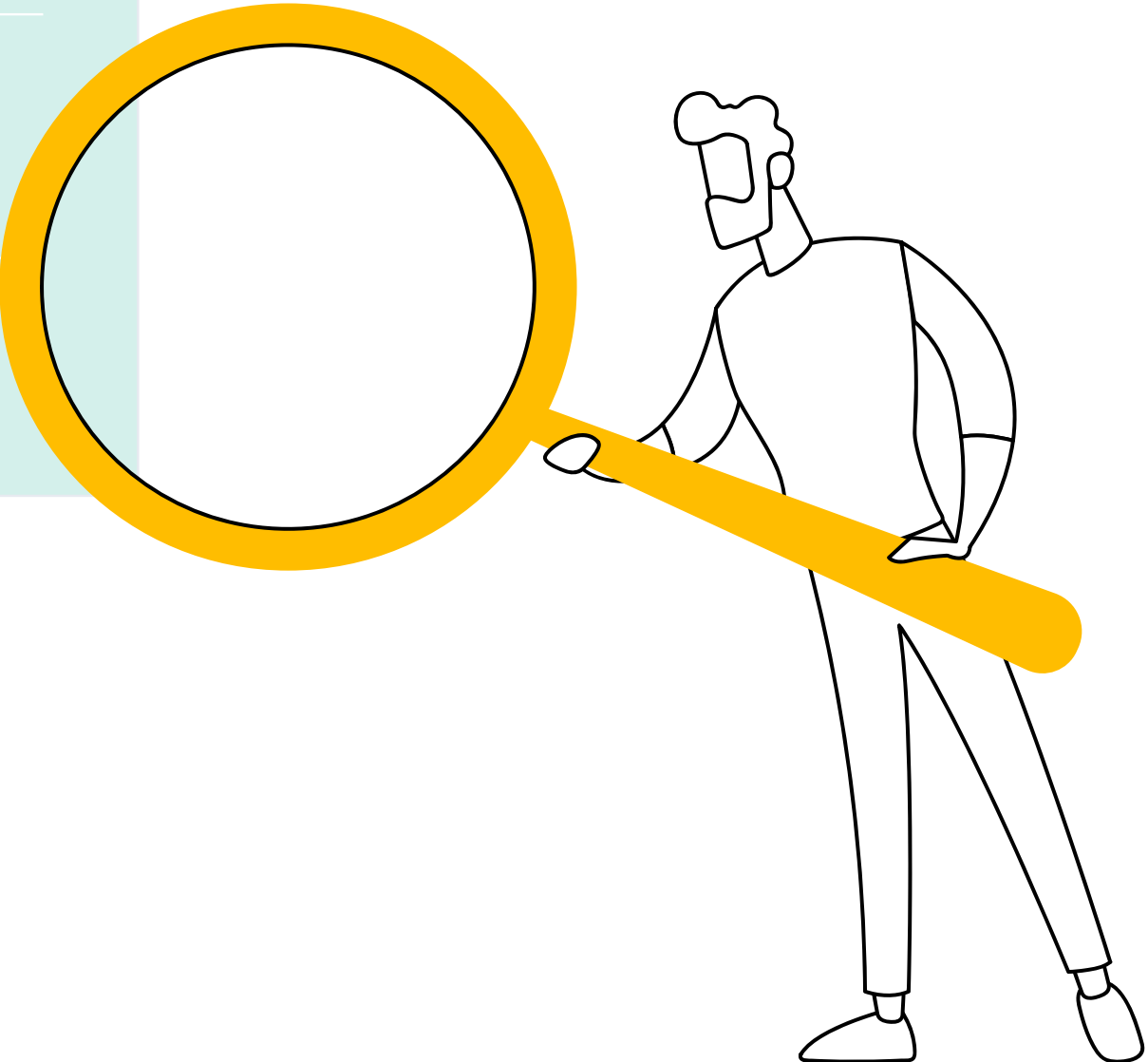
A one-time build with a defined payback period

Every workflow connected, built to work together from day one

Process built into the system, so it survives turnover

Owned IP on the balance sheet

None of this means spreadsheets, Zapier, or no-code tools are the wrong call. It means that once a business hits real complexity, custom software isn't the expensive option. Many times, it's the more affordable choice.



CUSTOMER RESULTS:

Before & After

The problems on the page before this aren't hypothetical. Here's what they looked like at three real businesses, different industries, same underlying pattern.



	Before	After
 Custom fabrication business	Job quotes lived in spreadsheets. Scheduling ran off a whiteboard, photographed daily to “sync up” with the shop floor. Quotes and actual costs drifted apart, and nobody saw capacity issues until a job was already behind.	Real-time job costing. Scheduling conflicts dropped sharply.
 Records processing business	Records split across a shared drive, a legacy database, and paper files only one employee could navigate. Simple requests took days, with no way to show a client where things stood.	Request turnaround dropped from days to same-day, with a full audit trail.
 Pool construction & service business	Quoting, scheduling, and change orders lived in three disconnected places. Change orders got lost, crews showed up to the wrong site, and margin wasn't known until a job closed.	Change orders stopped falling through the cracks. Margin visible while a job is still in progress, not after.

If you landed on Option 4, we'd like to talk.

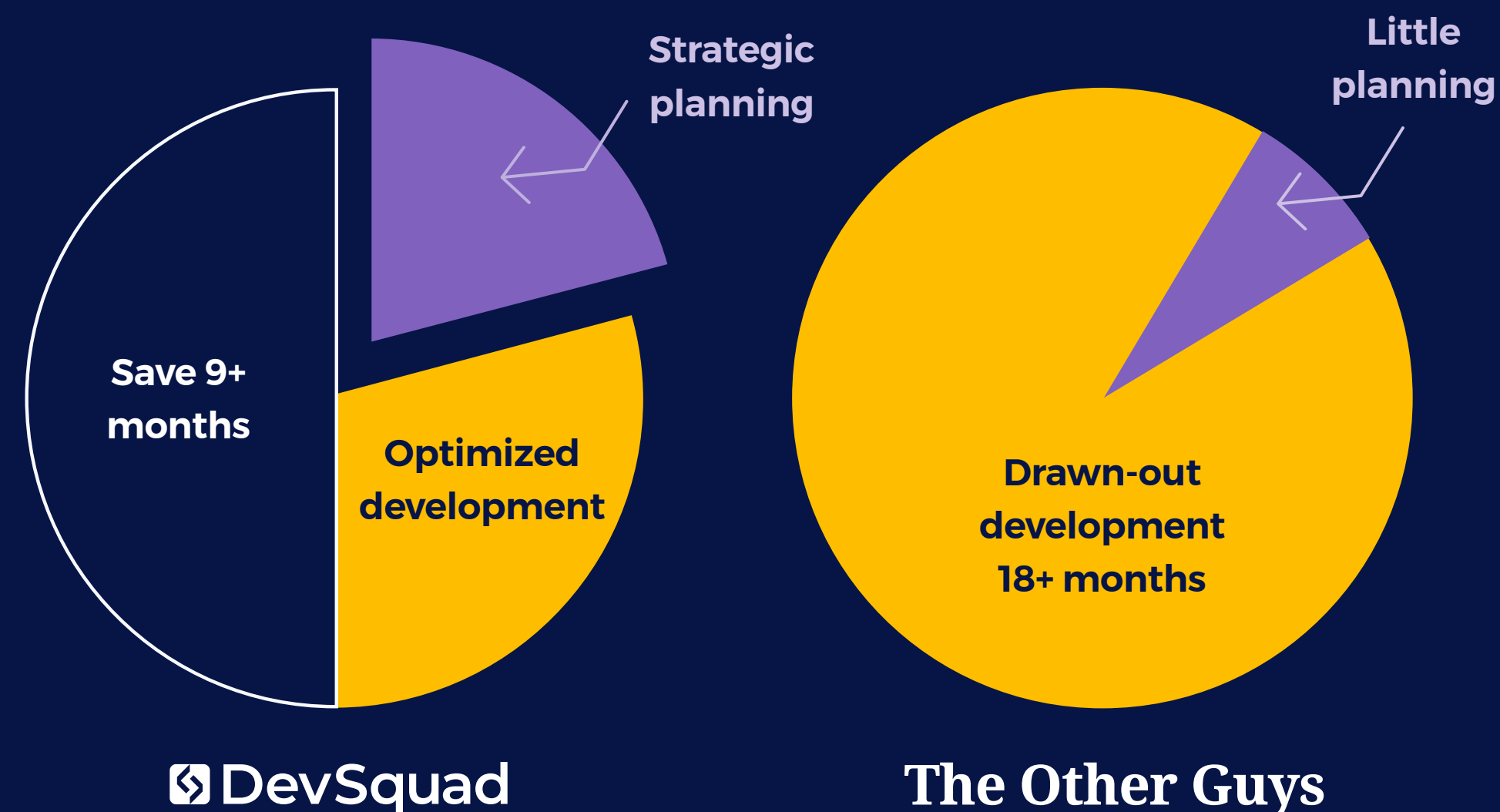
DevSquad builds custom software for businesses that have outgrown off-the-shelf tools, replacing spreadsheets and stitched-together stacks with a single system built around how you actually work. No pressure if you're not there yet. If you are, we'll start with a conversation, not a contract.

*"For custom internal software, the fastest path to value is often replacing what the business is already using today, whether that's an outdated system, manual spreadsheets, or expensive subscriptions. **Replace first, improve second, innovate third.**"*

— Mauricio Kiyama, VP of Product, DevSquad

Save 9 Months and 6 Figures with DevSquad

We devote more time to strategic planning and automate everything we can to deliver ROI faster.



[Schedule a call →](#)