

BUILD VS. BUY:

AN HONEST GUIDE TO DECIDING ON CUSTOM SOFTWARE



Build vs. Buy: An Honest Guide to Deciding on Custom Software

Custom software is powerful when it fits a problem no product solves — and expensive when it duplicates something you could have bought. Most bad software decisions aren't about code. They're about choosing to build when you should have bought, or buying when you should have built.

This guide gives you a plain framework to make that call before you spend a dollar: when off-the-shelf wins, when custom is worth it, the third option most people skip, and what actually drives the cost either way. No sales pitch — at QOS Software we tell clients to buy off-the-shelf whenever a proven product already does the job.

1. Start With the Problem, Not the Software

The build-vs-buy question is second. The first question is what problem you are solving and how much it is worth to solve it.

Define It First:

- What specific outcome do you need — and how do you measure it?
- Who uses this every day, and what slows them down now?
- Is this process a competitive advantage, or just plumbing?
- What happens to the business if you do nothing?

The Rule:

If you can't describe the problem in one sentence, you're not ready to choose a solution — custom or packaged.

2. When Off-the-Shelf Is the Right Call

Buying is usually the smarter default. A mature product spreads its development cost across thousands of customers, and someone else maintains it.

Buy When:

- The need is common — accounting, payroll, email, CRM, scheduling
- A proven product already matches how you work
- The process isn't unique to your business
- You want someone else to own updates and security patches
- You need it running in weeks, not months

The Bottom Line:

If a packaged tool does 90% of the job and the other 10% doesn't hurt, buy it. Building to avoid a small compromise is the most common way custom projects waste money.

3. When Custom Software Is Worth Building

Custom software earns its cost when the software itself is the advantage — not just a tool that supports the business.

Build When:

- Your workflow is unusual and no product fits it
- The process **is** your competitive edge and shapes how you win
- Disconnected systems need to share data cleanly
- You're paying to work around software instead of with it
- Per-seat licensing punishes you for growing
- You need to own the code, the data, and the roadmap

The Goal:

Build the part that makes you different. Buy everything else.

4. The Third Option Most People Skip

Build and buy aren't the only choices. Often the right answer is to configure a platform you already own or connect the tools you already run.

Consider Before You Build:

- **Configure** — many platforms flex further than teams realize before code is needed
- **Integrate** — a small connector between two products can remove the pain without a new system
- **Extend** — add a focused custom piece onto a product you keep, rather than replacing it

The Rule:

The cheapest custom software is the custom software you didn't have to write. A good partner looks for the smallest change that solves the problem.

5. What Actually Drives the Cost

No one can quote custom software from a headline. The price follows the scope — and a handful of drivers move it more than anything else.

The Levers That Matter:

- **Scope** — how many features, screens, and user roles the software must support
- **Integrations** — every external system you connect adds design, testing, and failure cases
- **Complexity** — simple forms are cheap; real-time, calculation-heavy, or high-volume logic is not
- **Data migration** — moving and cleaning existing data is often underestimated
- **Compliance** — regulated data raises the bar on security, testing, and documentation
- **Platforms** — web only is one cost; web plus iOS plus Android is three surfaces to build and maintain

The Honest Answer:

"It depends" is true, but incomplete. A good shop tells you **what** it depends on and helps you cut scope to fit a budget — instead of hiding the number.

6. The Costs Both Sides Forget

The build-vs-buy comparison usually stops at the first invoice. The real comparison runs for years.

Count These On Both Sides:

- Ongoing maintenance, hosting, and support — software is never “done”
- Subscription fees that rise every year, and per-seat costs as you grow
- The cost of workarounds when a bought tool almost fits
- The cost of being locked into a vendor you can’t leave
- Training, updates, and the time your team spends fighting the tool

The Rule:

Compare the five-year cost, not the first-month cost. Cheap-to-buy and cheap-to-own are not the same thing.

7. Questions to Ask Any Development Shop

Before you hire anyone to build, these questions separate a partner from a vendor.

Ask:

- Will you tell me if I should buy off-the-shelf instead?
- Who owns the source code and the data when we’re done — in writing?
- What drives the cost, and where can we cut scope to fit a budget?
- What happens after launch — who maintains it, and how is that priced?
- How will you integrate with the systems we already run?
- Can you show me something you built that’s still running years later?

The Red Flag:

A shop that never recommends buying, never mentions ownership, and never talks about maintenance is selling a project — not solving your problem.

8. Own What You Build

If you decide to build, ownership is the point. Software you don't own is just a subscription you paid to create.

Ownership Means:

- You hold the source code and the intellectual property outright
- You can change developers without rebuilding from scratch
- You control the roadmap — features ship on your priorities, not a vendor's
- Your data stays yours, in a form you can move

The Bottom Line:

Get ownership in the contract before the first line of code — not after.

The Final Word

Build vs. buy isn't a loyalty test. The right answer is whichever one solves your problem for the lowest total cost over time — and for most needs, that's buying a proven product.

Build when the software is your advantage. Buy when it isn't. Configure or integrate when a small change will do. And whoever you hire, make sure they'll tell you the truth about which one you actually need.

Next Step:

Write down the one process that costs you the most time or money today. Then ask: does a proven product already solve this? If yes, buy it. If no — and the process is part of what makes you competitive — that's where custom software earns its cost. QOS Software will help you tell the difference, honestly, before you spend anything. Call **877.800.7672** or visit **qossoftware.com**.