

# Fintech Build vs. Buy

There are no solutions. There are only trade-offs.

So wrote Thomas Sowell, the award-winning economist, when describing the fatal flaw of many well-intentioned plans: when we think we have the perfect solution, we blind ourselves to possible roadblocks, making us less likely to overcome them when they inevitably arise. By contrast, acknowledging the downsides that come with each alternative prepares us to make the best choice among the options that are actually available.

Along with countless other situations, this lesson applies to fintechs choosing whether to build their own software platform or buy an existing option from a vendor. Fintechs need a system that can support their products and processes, scale with growth, keep up with shifting compliance demands, and meet any number of needs specific to their business model. The question then becomes whether they can find an alternative to meet those needs, or if they need to build their own system.

---

There are no correct choices here; only trade-offs. But analyzing those trade-offs will help your company determine which option gives you the best chance of meeting your needs and accomplishing your goals. In this article, we'll cover four major trade-offs:

-  Control
-  Costs
-  Expertise
-  Risks and unknowns

And from the outset, we should acknowledge that this choice isn't binary. Some fintechs build a core system and then bolt on integrations to flesh it out, or build their own custom application on top of a vendor's platform. The hybrid approach doesn't bypass the trade-offs, but instead threads the line between them.

# Control and Responsibility

The broad appeal of building your own platform is the promise of complete control over the system—everything from the overall vision and development direction to the finer details of your tech stack and implementation. And yet, that appeal is also the main detriment to the DIY approach. Everything that needs to be built rests squarely on your shoulders.

Buying software from a vendor is a perfect inverse: less control, but also far less responsibility.

Here is a quick chart to compare what you would control (and thus be responsible for) in a build or buy approach:

|                         | BUY                                  | BUILD |
|-------------------------|--------------------------------------|-------|
| Access                  | No Access to Source Code or Hardware | 100%  |
| Control                 | Limited                              | 100%  |
| Ownership               | Data Only                            | 100%  |
| Overall Software Vision | 0%                                   | 100%  |
| Project Planning        | 0%                                   | 100%  |
| Software Expertise      | 0%                                   | 100%  |
| Technology Decisions    | 0%                                   | 100%  |
| Software Maintenance    | 0%                                   | 100%  |
| New Feature Development | 0%                                   | 100%  |
| Compliance              | 50%                                  | 100%  |
| Security                | 0%                                   | 100%  |
| Hosting                 | 0%                                   | 100%  |

Buying will always be the easier choice, but it doesn't offer you the same granular control that you get from building your system. If you need direct control of these items, you may need to build your own system to get it. But if you're comfortable simply owning your data within someone else's platform, then buying is more attractive.

## Costs

Cost, of course, is a limiting factor for both build and buy. Some fintechs suppose that building is like cooking at home, and buying is like eating out, with building being the obvious choice for a conscientious budgeter. This isn't always the case. To extend the metaphor, a bowl of cereal from your own kitchen will be less expensive than any restaurant dish, but if you're buying exotic ingredients and top-grade kitchen appliances, your home cooked meal can quickly outpace restaurant fare.

Here are some of the primary costs associated with buying or building software:

| <b>BUILD</b>             |  | <b>BUY</b>                        |
|--------------------------|--|-----------------------------------|
| Initial Development      | Ongoing Code Maintenance                         | Activation/Delivery/Configuration |
| Hardware                 | New Features Specifications                      | Subscription Fee                  |
| IT Support               | New Features Development                         | Usage Fees                        |
| Employee Training        | Software Resources (e.g. libraries, tools, etc.) | Employee Training                 |
| Ongoing Hosting/Hardware |  | Upgrades/Additional Modules       |

The specific cost of buying will depend on the vendor and the details of your contract, but in general, it will be far more predictable than building. That's true of both initial and ongoing costs, with subscription and usage fees remaining more or less steady while code maintenance and IT support costs can spike unexpectedly.

System resilience is often overlooked when considering long-term costs. When the system goes down, how long will it take to bring it back online or restore full performance, and how much money will be lost in the downtime? Builders will need to handle outages on their own, but buyers can be put in the difficult position of waiting helplessly while an unprepared vendor struggles to restore service.

While cost is far from the only consideration, it's an important one. Many companies have set out to save money through a custom build only to end up spending more on their own system. Whether you can predictably come in with lower costs will depend to a large extent on the next trade-off: expertise.

## Expertise

---

Building a platform requires expertise both in the field (finance) and the tools you'll use to create it (the specifics of software, hardware, and development). Here, the trade-offs will be different depending on whether you already have that expertise.

Building is more feasible for companies with expertise in both finance and tech. Some of the best builds have come from companies who were living the problems their software solves day to day. Even then, there is a trade-off: the opportunity costs of building something new should also be considered. Will your business be better off if you deploy the resources you'd dedicate to a build in some other way?

If your company lacks software expertise or finance experience, building will be an uphill battle. You can always hire on the talent you need, but significantly expanding headcount with high-skill roles will quickly inflate the project's cost. The more people you need to

hire on, the better option buying becomes in comparison.

Fintechs also need varying degrees of expertise in security and compliance. Hiring in these specialized domains is costly, but not as costly as the fines that the CFPB, OCC, FTC, or countless other federal and state agencies levy against companies who violate their regulations. What's more, today's political climate often sees radical shifts in policy with each election, meaning that compliance will be an ongoing concern and cost for any fintech. While builders will have to bear that compliance burden alone, buyers don't get off scot-free either. Some vendors have tools that can significantly reduce that burden, but ultimately every company handling customers' financial and personal data needs to ensure they handle that data in a compliant, secure way.

## Risks and Unknowns

---

So far, we've discussed several risks with building your own system. If you're missing the funding or expertise necessary to build a platform, you're likely to run into problems. Harvard Business Review found that one in six IT projects has an average cost overrun of 300% and a schedule overrun of almost 70%. McKinsey, meanwhile, found that 56% of IT projects fall short of their original vision. One estimate of IT failure rates is between 5% and 15%, which represents a loss of \$50 billion to \$150 billion per year in the United States.

But buying an existing product isn't foolproof either. Software vendors all excel in different areas. You might find one who meets your needs in product configuration, industry expertise, compliance, security, scalability, or other areas, but they might not excel in all of them. Among vendors, you'll have to analyze their strengths and weaknesses, determining which trade-offs you're willing to make.

## Conclusion

---

Building and buying both carry their own risks and rewards, stemming from the nature of the project. Building a platform from the ground up means far greater control, but it requires far more expertise to manage that responsibility. And while costs can come in lower, they could also balloon until they exceed the cost of simply buying. Buying from a vendor involves different kinds of risks. Costs and launch times will be much more predictable, but failing to properly align with a vendor can introduce variables to your business that you have no control over.

If you need total control over your operation and tech stack, building is the better option, but only if you have the expertise, funds, and time needed to execute on it. If that level of control isn't necessary, then buying is the better option. (And then begins the challenge of deciding which vendor best fits your needs.)

And what if you need total control over the operation, but are lacking the expertise, funds, or time? Unfortunately, there's probably not a perfect solution for you—only a trade-off.

# Start innovating with LoanPro today

Talk with our team about driving innovation for your organization today.

Let's Chat →