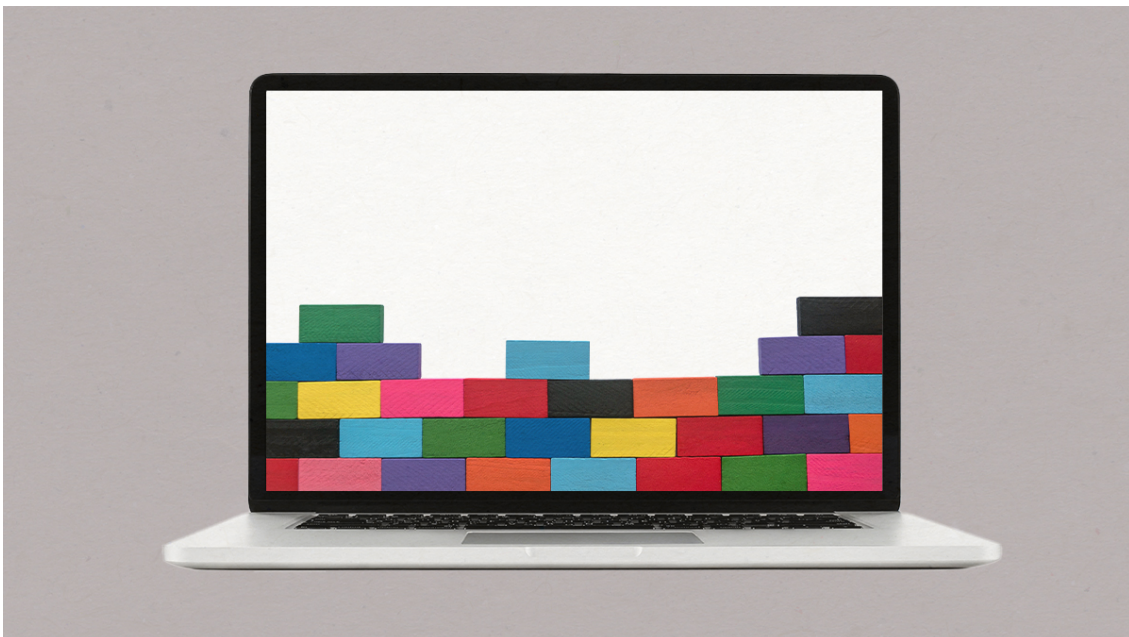


When Should Your Company Develop Its Own Software?

by Robert Sher

December 07, 2021



HBR Staff/Mohamad Faizal Bin Ramli/Mensent Photography/Getty Images

Summary. Leveraging homegrown software to bring innovation to your market or to create more efficient operations can be a strong growth driver. But the buy-it vs. build-it decision is a critical one. If buying the software you need just isn't possible, building it may make sense. But there's no denying that's a difficult path, and only worth it if the upside is big. Before you build, make sure you understand the real costs to succeed over the long term, and only embark on those code-writing efforts you're sure your business is capable of. The author discusses two midsize companies that successfully "rolled their own code" and presents three competencies required to do so. [close](#)

Every company needs and uses software, and some is a significant driver of business success. But as small companies grow to midsize, software performance gaps can emerge. Finding new software solutions can fix problems and inefficiencies and help teams develop innovative products and services. But midsize company CEOs often face a difficult choice: whether to upgrade through a vendor or develop (a.k.a. “roll”) their own code.

It’s widely understood that software upgrades are always expensive and often disruptive. Sometimes they fail completely, or they don’t deliver on their original promise. That means little or no return on money spent. But sometimes, there’s simply no off-the-shelf software available to address a business’s unique problem.

For small companies, it’s usually easier (and almost always cheaper) to do manual workarounds when their operating software isn’t up to the task. But midsize companies can lose a great deal of money and stunt their growth due to the inefficiencies that inevitably spring from such workarounds. And those tortured manual processes can prevent companies from seizing opportunities in a timely manner. For those companies, custom coding is a viable option. (Large businesses with deep pockets can build software development teams and often have the talent on board to do so.)

Most midsize companies have a “super user” who’s good at helping everyone with the capabilities already built into their software (like report writers, dashboards, etc.). And most modern enterprise resource planning (ERP) software has layers that allow for customization — often a layer where value-added resellers (VARs) can make changes and a customer layer for customer customizations. If a midsize company can get what it needs from that, fantastic. But what if it can’t?

Many midsize companies get stuck trying to decide whether to buy new software or attempt to write their own code, even if that just means connecting disparate systems. Others try to outsource the problem to a software firm. While outsourcing code creation

may be part of a solution, doing so successfully requires rigorous project management — a capability not all midsize businesses have.

Meanwhile, the clock is always ticking. Efficiencies that could be realized with software aren't retrieved, eating away at margins. Market opportunities are lost to competitors. How can midsize business leaders determine when it makes sense to build their own software?

When to Roll Your Own Code

It's inefficient to develop custom programs for core business functions like accounting, payroll, sales tax, inventory, and customer relationship management (CRM), and so many options are readily available. But if there's no software that does what you need it to do, you may have no choice but to roll your own, especially if there's a high-value opportunity to seize or a significant efficiency to gain. (Creating your own code is only worthwhile if there's a big payoff; without a strong ROI, forget about it.)

For example, in 2007, BF&S Manufacturing was gaining steam as a contract manufacturer for complex, low-volume — but critical — components for aerospace, military, medical, and industrial verticals. Its customers wanted to oversee the work, but BF&S was based in Mexico, and many of its customers didn't want to invest the time and money to travel and stay there.

BF&S depended on a close relationship with its customers, often turning to their engineers to solve production problems. But distance and a border were making that ever more difficult. Screen-sharing and cameras alone weren't going to be enough for its customers, and BF&S feared losing them to more closely based manufacturers, even if those businesses charged more. BF&S needed to be able to port valuable production data from its core ERP system into a format its customers could use.

BF&S CEO Carlos Fernandez looked around but couldn't find a solution to buy. Instead, he says, "We embarked on a software program that would provide 24/7 real-time data" on the company's product builds. It started with their "computer guy," as Fernandez calls him, just out of college, building a tool to track raw materials, work-in-progress, and finished goods inventories and provide visibility internally and externally.

It was completed and first used in 2010. Customers loved it. Fernandez began to grow the software development team in Mexico, supporting four facilities in the state of Sonora with a combined headcount of 500. Customers could now see video of the workstations, their products' progress at each step, BF&S's raw and finished goods inventories, who was working on their job, and all the product stories and specs.

This custom coding required a keen understanding of both the company's business and its customers' needs. Originally headed by Fernandez, the team of engineers and operations leaders now plan and manage the ongoing support and development of the tool.

Today, although Fernandez won't claim that his company's home-built code is a huge competitive differentiator, he believes it gives his customers what they want and what he couldn't provide through off-the-shelf software: transparency into and a measure of control over the production of their products.

The Journey and the Costs

Rolling your own code is neither simple nor cheap. Software engineers are highly paid. In the United States, that means six-figure salaries. The costs of finding and hiring engineers often involves search firms, which charge 15% to 30% of the first year's salary, and for the past several years, even they've been struggling to find good candidates. On top of sourcing costs, you must interview and assess candidates for technical skills, train and onboard new hires, and provide a digital environment for development and testing.

And then you have to manage the code development tasks, making sure they're productive. As the development department surpasses five or six engineers, you'll need a DevOps executive to supervise it — if programmers are undermanaged, days and weeks can be lost while productivity plummets.

And you can't just hire developers and managers and expect the magic to happen. Engineers make what the business tells them to make. They thrive on clarity. So, you're going to need to spend time getting your arms around your business's opportunities and needs to be able to describe the features, functions, and options you want. That software roadmap must be completed before your engineers start coding. Fail to do all this well and on time, and you'll have very expensive talent sitting on their hands, likely looking for other places to work.

Finally, when you develop custom code, you need to maintain it. Software breaks down all the time. Hackers continually find new attack vectors. New needs pop up and users demand modifications. Even programming languages age, so every five to 10 years, software may need to be rewritten. The costs keep coming.

However, while custom coding is challenging, it can be a pivotal factor and well worth the trouble for some companies that are innovating solutions for their customers.

Corefact (a Mastering Midsized client) is a full-service marketing services provider for the real estate and mortgage industries. In 2005, the company came up with a fresh idea. If a realtor could send a postcard to a potential client with a unique URL that would take the client to a website with their own home at its center, that could be hugely appealing, and a possible game-changer. Corefact's customers, realtors, were excited, not only by the potential appeal to their clients, but also by all the data this kind of engagement would supply them with.

Corefact couldn't buy software to do this — it was new. Corefact's founder and CEO Chris Burnley had always been a technologist. Prior to Corefact, he started several technology-driven

companies. Thanks to this technological competency, the company found a way to print variable data — unique URLs — on postcards and then move them on to web servers that would wait for a homeowner to type in the URL, after which a new, unique website would be created instantly. By 2006, the software was launched with a single engineer.

Today the engineering team has grown to 10, located in the U.S. and abroad. They've created custom code that's not only customer facing, but that also efficiently brings together thousands of daily orders through order entry, graphics, and pre-press and automates the efficient flow of work onto presses and through finishing.

Burnley says, "Our original concept put us on a fast ramp for growth, but our ability to innovate with technology continues to propel us. Of course, the investment in engineers is huge and ongoing, but the list of opportunities is long."

But they don't build every piece of software they use. When it came to upgrading their ERP, they chose a standard product by Netsuite, into which they're connecting their self-made order-handling systems. Similarly, they've recently dropped a self-made CRM in favor of Salesforce, keeping their development team focused on creating software they can't buy.

The Three Competencies You Need to Roll Your Own

The examples I've discussed require different amounts of the following three competencies, depending on how complex your custom code requirements are:

Translating business needs into software projects. Identifying business needs — and their solutions — is a necessarily iterative process, keeping in mind the limitations of existing software, as well as your resources and available data. This is neither software development nor business management; it's a form of engineering where one leg stands in the business and the other in a thorough understanding of how your current software systems work.

This competency could be held by one executive in a smaller midsize company, or by a small team as the organization grows. What goes in is a problem or opportunity, what comes out is a series of detailed steps to create and maintain code: exactly what data is to be used and what logic or processes should be used to produce a solution. Without all these steps, endeavoring to create custom code makes no sense.

Code development. Depending on the circumstances, a midsize business could have one programmer or a full engineering department. For example, at my prior company, we had Dave, a young warehouse employee who coded as a hobby, come upstairs now and then for small coding projects. For bigger opportunities, code development can grow into a series of engineering teams with different skills and focuses working in a complete DevOps department, led by a VP or chief technology officer.

Software operations. The operations side of managing custom applications is expensive — you need to maintain the health of the custom code and make sure your processes, people, and tools are kept up to date. Elements of operations include user support/help desks, training, security risk management, bug fixing, ongoing additional customization, uptime and performance attributes, and more.

Leveraging homegrown software to bring innovation to your market or to create more efficient operations can be a strong growth driver. But the buy-it vs. build-it decision is a critical one. If buying the software you need just isn't possible, building it may make sense. But there's no denying that's a difficult path, and only worth it if the upside is big. Before you build, make sure you understand the real costs to succeed over the long term, and only embark on those code-writing efforts you're sure your business is capable of.

RS

Robert Sher leads Mastering Midsized, a

consulting firm that partners with leaders of midsized companies to build strong business infrastructure needed to grow predictably and sustainably. His most recent book is *Driving Midsized Growth: People*. He can be reached at R.Sher@MasteringMidsized.com and [@RobertSher](https://www.instagram.com/RobertSher).