Build vs Buy

Examining the Options for Your Future
Customer Engagement Platform

COCONUTSOFTWARE.COM

Introduction

Financial services organizations looking to implement technology based solutions to improve customer experience, staff operations, data collection and other areas of their business have more options than ever these days.

With the increasing demand from both organizations and their customers for a frictionless and personalized user experience, third party offerings are becoming a more interesting option for financial institutions. It's easier than ever for companies to find a software provider with a solution to fit their needs. At the same time, this demand is creating an ever expanding pool of talented designers and developers, allowing organizations to bring in skilled professionals to strengthen their in-house capabilities and build their own custom solutions.

CONTENTS:

Cost & Control

1.0

For most organizations looking at the question of build versus buy, it typically comes down to two important factors: cost and control. The goal of minimizing cost while maintaining as much control as possible plays a major role for nearly every company looking to implement a new software solution.

With these key factors in mind, it seems as though the obvious choice would be to build your software in house. After all, building allows you to customize your solution and adapt it to your specific needs, rather than trying to adapt to one that was built for a general market. At the same time, you could plan and control the level of investment by utilizing your own in-house developers. But it's rarely quite as straightforward.

In this report we will examine the critical questions financial institutions should ask when looking at the 'build or buy' issue, examining the pros and cons of each route to allow for an informed decision.

CONTENTS:

Is it less expensive to build your own solution?

While there are most definitely occasions when building your own software could save you money, the unavoidable truth is that this route has much higher potential to cost you more than you would have hoped. Between rework, delays and inexperienced resources learning and testing as they go, the costs can add up. Software development is generally a lot more involved and complex than non-SaaS companies believe, with hidden costs, risks and complexities that only often show themselves once a project has been committed to and the code is starting to be written.

According to a report by Standish Group, 19% of large IT projects are scrapped completely after spending money, having investing resources and time in the project. Additionally, according to an article by McKinsey, "On average, large IT projects run 45% over budget and 7% over time, while delivering 56% less value than predicted." If (or rather when) your team encounters these issues, you then have another choice to make – do you continue with development and battle your way through? Do you change the scope of the project? Or do you cancel it all together? The sunk cost fallacy often results in future delays and investments, purely due to the perceived time and effort already invested.

"Next thing you know, you've spent literally hundreds of hours building a tool that's not core to your business. Hours that really hardly saved you any money. But worse than that, it was hours that weren't spent making more money by improving your core product. Not only did you save an insignificant amount of cash, you actually stifled future cash."

Josh Pigford
Founder, Baremetrics

Cost: Factors to Consider

BUILD

As it's an internal project, your organization can decide how much it wants to spend on such a project.

While the upfront cost of building a custom software solution may be higher than buying one, the long-term cost of implementation may be lower (for example, with no ongoing software license fees to pay). Most large financial institutions will have the IT resources in house to work on a customer engagement platform, but there would likely be delays due to inexperience. Consider also the costs of having those individuals not focusing on their primary roles and the ramifications of backfilling.

Delays in rolling out the newly built in house platform will have impacts on total project costs, but will also affect your revenue through missed engagements, continuation of the status quo customer experience, and the slowed resolution of the challenges that kicked off the project in the first place.

In-house IT projects are well known for experiencing major problems and costing far more than the estimated amount.

The temptation to reduce costs could lead to corners being cut and a solution that's not fit-for-purpose, as opposed to a specialized vendor solution which fits your needs 'out of the box.'

Ongoing maintenance of a custom built, in-house solution can be considerable with significant implications on customer and staff experience. Disconnected systems built in silos result in orphaned data, multiple channels to manage and update as well as code duplication, complex testing environments and prolonged release cycles.

BUY

Building your own solution in-house requires you to build a team to do it. Good software developers don't come cheap.

Commercial solutions are usually much faster to implement as they have been purpose built to scale quickly with repeatable processes and tested change management methods. These companies also tend to employ specialists who have deep subject matter expertise on topics like scheduling, visitor and queue management, contact center engagements and customer journey mapping.

When you buy a solution from a third party, that vendor bares the costs of keeping the platform up to date and is accountable for regular code releases and fixes, as well as designing a future proofed product roadmap.

The right vendor also has a vested interest in understanding the financial services industry to ensure that the solution they are building, not only for today but also well into the future, is going to address the changing needs of their current and prospective customers. Buying a solution from a vendor that specializes in the financial industry means they'll understand the typical workflows and use cases, as well as procurement and implementation processes that work most effectively.

Does building provide more control?

If you build your own software, you own it. You get to control exactly what it does and how it looks. You can make changes whenever you need to. You have complete control. Of course, it's not quite so simple.

The truth is that whether you build in-house or buy a third party solution, to a certain extent both methods leave you reliant on the people who built it. So, while you don't get to control every aspect of a third party solution, if a key member of your development team ever leaves to pursue a new opportunity, you could discover their code was not written in a way that allows a new member to edit or interpret it.

At the same time, companies actually have a lot more influence on vendors than you would expect. You won't 'own' the source code, but you will be able to provide feedback, suggest changes and have a say on updates, features and releases. This ability comes down to who you work with, and not every vendor is going to be able to provide the same level of control and input from their clients.

"Software vendors have to be much more responsive to customers than they have in the past. Those vendors that ignore customer feedback and simply march to their own tune will not survive."

RL Solutions

Control:
Factors to Consider

BUILD

If you build it, you own it. You don't have to worry about competing with other customers for a vendor's attention when it comes to creating tickets for bug fixes or requesting changes.

Building a custom software solution in-house gives you the freedom to choose which third-party applications you integrate with, whereas some vendor solutions may be more limited in terms of API integration. This is limited by the scope of expertise of your in house talent.

Depending on how you want to grow, a custom-built solution may give you more scalability than a rigid system which isn't built with flexibility in mind.

You may not be 'locked-in' to a vendor by choosing to build it yourself, but you'll be reliant on your in-house developers when it comes to maintaining, fixing or upgrading your in-house system.

If you build your own solution, will you have the ongoing resources available to maintain and improve it?

BUY

Competition between vendors to win your business means they are constantly investing in improving their software and meeting the demands of their prospects and customers.

Do you know what your competitors are doing? While you may be stuck in development purgatory trying to build your own software, the competition may be buying specialized off-the-shelf solutions that provide faster implementation, better results and competitive advantage.

Choosing a vendor that is receptive to feedback and is deeply integrated in the financial institution space means you'll have a direct hand in the future development of the solution you're buying.

Asking the Right Questions

2.0

If after looking at the pros and cons you decide to go the build route, there are a few questions you need to ask yourself before committing to the project.

CONTENTS:

What do you really need in a customer engagement platform?

Before you can properly examine your options, you need to pin down exactly what it is you need your solution to do - and there's a big difference between what you need and what's nice to have. The allure of building software is that all requirements can be satisfied, but unfortunately, that's a fantasy. Resource constraints mean coding must be prioritized and some requirements will never be met; or the team may not fully understand the problem domain, and may not discover unknown requirements. Scope creep is a real issue, especially in a large, complex environment like an enterprise financial institution.

That's why it's vital to map out the problem you're trying to solve. Examine it closely and work backwards to identify how it needs to work to fit in with your existing systems. The more it needs to integrate with, the more complex its development is going to be. Validate your assumptions and decisions with internal and external stakeholders to ensure you aren't creating in a vacuum.

For a piece of software that doesn't have any effect on other areas of your business, or for ad-hoc applications specific to a single business process, building could be a good choice. But when your solution needs to integrate with multiple departments, branches, specializations and touchpoints, it's almost guaranteed that your custom solution is going to cost more than a third party solution.

On Hubspot's decision to abandon their internal build and partner with a vendor;

"Ultimately I think there's no solution that will give you everything you need for every single analysis you have to do... We thought we were a special snowflake and things we had to have were mission critical."

Daniel Wolchonok
Head of Product Analytics, Hubspot

How much will it cost, both direct costs and lost opportunities?

Calculating the upfront investment you'll devote to a piece of custom built software is just the beginning. On top of the direct costs of buying or subscribing to an existing tool versus developing a new product, you also need to consider the time-savings and opportunity costs that come with both.

UPFRONT COST

For in-house solutions, pricing is rarely predictable. But one area where they show consistency is that they regularly cost more than anticipated. According to a McKinsey survey of IT executives, large IT projects run over budget 45% of the time, while delivering 56% less value than planned. The same study found that 17% of projects go so bad that they "threaten the very existence of the company."

To make a rough guess on the final cost of a complex custom solution, a good rule of thumb is to multiply the cost of the off-the-shelf solution implementation by 10. That means that a $10K off-the-shelf system will cost $100k to develop custom capabilities, or customizing a $100k system would cost $1M. Ten times the price is a significant difference. Is complete control over the software worth the price? That's up to you.

OPPORTUNITY COST

If you allocate existing resources to building and maintaining this software, what will you have to give up? Where are you taking resources away from? Effective in-house solutions typically require a dedicated team, meaning you'll have to permanently move developer talent away from current projects or hire additional headcount. To assess your opportunity cost, consider the best use of that talent and where your top developers can have the highest impact.

Regardless of the size of your company or amount of talent on your team, your resources are finite – and you have to make careful decisions about how you utilize them. Building your own solution comes with plenty of opportunities for delays, bugs, security vulnerabilities, etc. The more issues you encounter, the more precious time and resources you tie up working to fix them.

"Everybody knows that the more you buy off-the-shelf, the more cost effective it will be for both implementation and ongoing maintenance."

Mark Lutchen
Former Global CIO, PricewaterhouseCoopers (PWC)

Can you manage and mitigate the risks?

STAFF

Talented developers are in high demand. If a vital team member who worked on your custom software left to pursue a new opportunity, would you be able to make changes to (or even maintain) it without them?

SCALABILITY

As a business grows and changes, custom code designed for one purpose can become difficult to fit into the new environment. Will your team be able to write your software in a way that allows it to scale in the future?

SECURITY

All software development projects face security concerns, and regulations around the financial industry are even higher. Can your internal team ensure that all security protocols are met and maintained?

PROJECT FAILURE

According to Gallup, IT projects had an average overrun of 27%, one in six had an average cost overrun of 200%, and a schedule overrun of almost 70%. Even worse, 75% of IT executives admit that their projects are either always or usually “doomed right from the start”. While this doesn’t mean your project will fail, are you prepared to accept the risk of investing time, money and resources in a project that could?

COMPETITION

While your team is developing your custom software, competitors may be partnering with vendors that provide them with new competitive advantages in the customer experience. Moreover, some features (such as Reserve with Google) require partnership with a vendor. Is your solution going to deliver enough of a unique benefit to your customers to make it worth the risks involved in an internal build?

Conclusion

It’s almost always better to buy.

When going the buy route, the same questions need to be taken into account. However, finding the answers can be a lot easier. Once you’ve determined what it is you need, finding a vendor to partner with in providing it can be confidently secured. Issues with staff and project failure can be eliminated altogether, while other questions of scalability and security can be answered by asking the right questions and doing sufficient research into the options available.

For the vast majority of businesses, the risks and cost that come with building custom software consistently outweigh those of buying an existing tool. That’s because the effort of creating and maintaining a tool in-house takes away time, energy, and development resources from improving your core product and serving your customers.

The fact is, unless software development is the focus of your business (and sometimes even if it is), it’s almost always going to be better to buy when you have the option. You’re a financial institution, not a SaaS provider - play to your strengths.

> ***“I take the build vs. buy decision very seriously. Anything we build will have a maintenance cost in the future that has to be considered. Moreover, when you begin a project, the software that you are “going to build” always looks better than the software someone else already has because you haven’t yet run into the limitations that inevitably show up in software engineering. As such, we will buy wherever we can.”***
>
> *Timothy Campos*
> *CIO, Facebook*

Get up and running in only a few weeks.

COCONUTSOFTWARE.COM **+1 (888) 257-1309**

SCHEDULE A CONSULTATION

About Coconut Software

Our cloud-based, enterprise level appointment scheduling and visitor management platform combines your on-location and digital channels, providing self-service booking and curbside pickup, online and physical queuing, wait time tracking, callback requests and virtual meeting support as well as contact center integration.