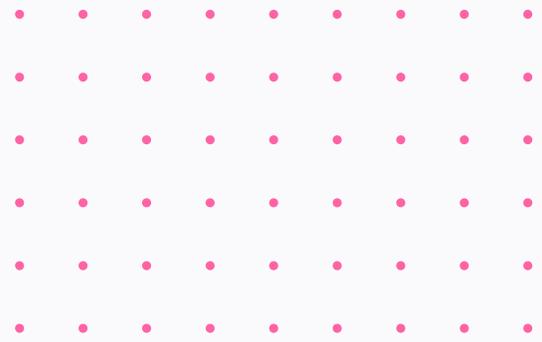# Buying vs. Building Resource Management Software

mosaic

You've all agreed that your organization desperately needs resource management software. But the pivotal question remains: should you build your own software or buy it? There are a lot of advantages and disadvantages to consider for both building and buying. On the one hand, buying software off-the-shelf can be quick, easy, and cheap. On the other hand, designing custom-built software can give you complete control over the features, integration, and code quality.

It's essential to do your research before making a final decision. Taking the time to consider different options and collecting feedback from key stakeholders is crucial before making a final commitment. Only by looking at all possible solutions can you make an informed decision that aligns with your unique needs and situation.

Weigh the time, cost, and resources associated with each option, as well as any functional integrations one or the other may lack. If you're serious about building your own software, be sure to speak to people who have gone through the process and built proprietary software. So which option is right for you? Let's take a look at the pros and cons of building and buying resource management software.

Benefits of building your own resource management software

Building internal resource management software can help your business in many ways. Tailoring the software to the individual needs of your organization can provide you with control over the process and influence over the timeline and costs associated with its development, making it easier to adjust resources where needed.

Creating simple, in-house internal software may save you money because you won't have to build features you don't need. Over a long period of time, building software can often be less costly than purchasing an off-the-shelf product, similar to how purchasing a car (versus leasing it) pays off the longer you drive it (though you with an outdated vehicle—or, in our case, outdated software).

Ultimately, it's important to carefully consider your organization's needs, time constraints, and cost factors when making this decision.

Challenge of building your own resource management software

When getting started, finding the right people to build and design your software can be difficult. If your organization doesn't currently build its own software, hiring people with the appropriate expertise may prove challenging, and hiring the best software designers and developers will be expensive.

Due to a lack of access to top designers, internal software teams often struggle to understand resource management problems and design appropriate solutions. This can raise costs and cause delays. Additionally, having multiple stakeholders inputting their own specifications can result in a "Frankenstein" product.

With a small development team, the transfer of information on the codebase can be a concern, especially when dealing with software involving multiple programming languages. It's essential to have a plan in place should the small team of people who built the software leave your organization.

Plan out all expenses

Cost is undoubtedly the single most important factor to consider when building your own resource management software, as the process can be very expensive. Depending on its complexity, tailored internal software can cost hundreds of thousands to millions of dollars.

When drawing up an initial budget, calculate the cost of all labor involved, from design and development to security and maintenance. In addition to software developers' time, don't forget to factor in the time that all stakeholders will spend on software design during all phases of the project, including the executives who will play a role in the design and testing. This is a significant and often overlooked expense that can be a large distraction from their core job functions, consuming time and resources that could otherwise be allocated to other projects.

There are also third-party software component licensing costs to consider. Integrating your current software into the custom software you are building can be a project costing tens to hundreds of thousands of dollars in itself, once you factor in time costs. Along with these considerations, ongoing maintenance and support are other important costs of building resource management software.

On a limited budget, achieving the code quality needed for reliable software may be challenging, but regardless, whatever you build will most likely encounter issues that require fixing. A quality assurance team is often required to ensure the reliability of your software, especially since it's mission critical. Additionally, there will be required updates and upgrades to open-source systems used in most software, not to mention training and the help and support provided to people using the software.

Furthermore, the software can be prohibitively expensive for small and medium-sized businesses that are unable to achieve an economy of scale for servers and other recurring expenses. For any organization, building your own resource management software will most likely be a costly proposition. All of these factors contribute to the high rate of failure of internal software projects.

Don't forget you will need to implement your security features. In an era of advanced cyberattacks, it's more important than ever to ensure that your security measures are up to date and effective. If your organization does work for clients that require higher security now or may do so in the future, you would want the software to have SOC II Type 2 compliance at a minimum to ensure the right process and testing is in place. These levels of security are complex, time-consuming, and very expensive for a single organization.

If you are a multinational organization, localization of the software for date, language, and currency can be expensive and time-consuming. There are many details like these that are hard to anticipate at the onset of a project and must not be overlooked.

While the list above isn't exhaustive, it's crucial to think beyond base expenses and anticipate additional costs related to delays, unknowns, the unexpected, scope creep, and troubleshooting. If you don't regularly perform this type of software development work, you could be drastically underestimating the time and cost it requires.

Even if you're just building a supercharged spreadsheet, comparing the time cost of the employees involved in the development can prove to be expensive. This expense can be quickly evaluated by comparing the annual cost of the software to your average cost rate (including overhead) for the total time estimated for the people involved in the project.

According to Glassdoor, the average salary for a software engineer in 2023 is $106,227, with top engineers earning well over $300,000. Depending on the size of your organization, the annual expense for purchasing software will often be lower than the annual salary of the engineers required. The involvement of executives in the organization in the project is where things can get expensive, especially when considering the opportunity cost of their time.

Scope creeps

Keep in mind that the process of building anything usually starts out simple and then grows in complexity. As any experienced project manager will tell you, whether it's construction or software projects, time can start adding up quickly, and initial budgets can grow to multiples of the original estimates.

Given how quickly costs can accumulate, it's essential for businesses to have a clear sense of their budget at the outset to ensure that their custom-built resource management software does not break the bank. Only by appropriately budgeting the entire life cycle of the project can an organization manage the cost of building its own software.

In many cases, the investment will be well worth the effort. However, what started as an effort to save money can quickly deteriorate into a significant business loss once you start factoring in the time everyone is spending.

Beware of anyone you hire who tells you that building the software will be easy or straightforward. They may be inexperienced or simply hold an optimistic view because they are not fully acquainted with the complexities. Further, external consulting software developers you are looking to hire will also benefit from the ongoing expenses described above. For reasons highlighted in this article, you may become stuck with them. They know that it will be too late for you to turn back because of how far in you are. The "Sunk Cost Fallacy" is primarily to blame for this.

85% of software projects fail

Studies have shown that up to 85% of internal software development projects fail to deliver a return on investment. It's the unknown unknowns that get people in trouble. If your business doesn't build software as a service, and you don't have anyone in leadership who can answer the question, "What are the best programming languages?" this could be a sign that you are headed down the wrong path.

Failure can often be the result of losing stakeholders, having too many decision-makers reluctant to take critical action, and running out of money due to delays and unforeseen expenses. When projects run long, and costs become a major factor, project stakeholders often begin to lose interest in seeing the project through to completion.

The pros of buying software

Buying off the shelf may be a more appealing option to you due to its associated cost savings and the promise of immediate implementation. The time cost of not having the product immediately available is often overlooked when attempting to build your own software, especially given the delays that arise as scope creeps, and the software continues to evolve due to unforeseen changes.

With an off-the-shelf product, you'll be buying professionally designed, flexible software that has been developed by experienced professionals to work for a wide range of businesses. Successful software companies typically receive and implement feedback from thousands of businesses like yours to design and build a drastically better product. If you understand how deep it goes, resource management is as complex as software can get.

Remember, no software will ever be everything to everyone, and looking for the perfect solution will lead to you never buying software or being disappointed. It's important to remember that if there was an all-in-one solution that did everything, it wouldn't do anything very well.
Building a scalable resource management system requires specialized knowledge and expertise in systems architecture. If the organization's in-house team lacks experience in designing and implementing scalable solutions, they may struggle to create a system that can handle increased demands and complexities as the company grows.

As the system does become more complex with added features and functionalities, maintaining and troubleshooting it can become increasingly challenging for the internal team. it will ahe to become its own company.

Given software companies' large-scale servicing of hundreds or thousands of businesses, integrations with your existing software may be available at a steep discount relative to building them yourself. A lack of knowledge or understanding about integrations may lead the team to underestimate the challenges or complexities involved. In many ways, they are the most challenging part, given dealing with the APIs of financial software and getting reliable integrations is highly complex software development work.

Depending on the software, you can access the latest, and typically very expensive, machine learning and AI technology. These cutting-edge features require deep domain knowledge outside most software developers' expertise and beyond the reach of private companies, as these engineers will seek work at AI software companies.

Not to be understated, by purchasing software, you also won't have to worry about the headaches and expenses of training, help desk support, documentation, videos, and ongoing maintenance, which are typically included in your software subscription costs. Due to the time cost of the entire user base of the company, these often-overlooked expenses can eclipse the cost of software development.

Making a decision

Pros of building software

- You'll have the flexibility to customize the software to your business's unique requirements.

Cons of building software

- You may not have the relevant experience to know the project's requirements or the technical expertise to build it in the right way.
- Building custom resource management software costs hundreds of thousands to millions of dollars over the life cycle of the software.

The ability to customize technology based on the specific needs of your business may be worth the additional work and time involved in building instead of buying, but you'll need to evaluate the total cost of ownership for each scenario beforehand. If you do decide to build your own software, be sure to prepare a thorough budget that accounts for design, development, integrations, servers, licensing, training, maintenance, and ongoing support.

When considering the costs of building your own resource planning and management software, it is essential to take into account upfront investments in development, as well as ongoing maintenance costs. Budgets need to be established with contingencies for scope creep and unknowns that can lead to delays along the way, and it's important to include support, licensing, and maintenance expenses so that budgets stay within established expectations.

Pros of buying software

- - A much faster time-to-implementation.
- - Significantly less money to invest up-front.

Cons of buying software

- - You may not get the features exactly the way you feel you need them.

If you decide to buy software, be sure to pick a partner that specializes in resource management and understands your specific needs. Check that they have excellent customer support and are willing to listen to your challenges. Look for flexible, customizable software that allows you to grow with it.

But what's more effective?

We've looked at cost and challenges, but internally developed software also typically lacks the same level of polish, potentially leading to a suboptimal user experience for employees. Usage is everything – what is the point of building software no one uses?

As Forbes highlights, "In most situations, working with a reputable provider to deploy a proven solution will allow companies to get the technology into the hands of users more quickly and generally ensures that the solution will have more comprehensive support throughout its life cycle...Why spend countless hours having your best people—or hiring new people—architect a solution that already exists and is proven in the market? Generally, a vendor has already solved the same problem hundreds of times, therefore bringing clients the benefits of best practices based on others' experiences."

People—the highest expense in most organizations—are being managed very inefficiently as a result of little to no visibility at the organization level. This results in a tremendous amount of wasted time and money. Resource planning and management are the core drivers of efficiency in managing people and can dramatically impact the profitability of any organization. Given the importance of resource management in the organization, this discussion should be about selecting the most effective tool for the job, not the cheapest.

Looking at it this way, building your own in-house resource management software could never compete with a specialty-purpose product. You wouldn't think to build your own project management software, but for some reason, people are fooled into thinking it's easy to build resource management software—arguably one of the most complex business functions.

If you do decide to buy instead of building your own software, Mosaic is building the most advanced resource planning and management software ever. For over a decade, we've researched requirements by working closely with businesses like yours to build the fastest, most customizable resource management software on the market. With Mosaic, you get the collective knowledge and resource management best practices of thousands of organizations globally.

Have ideas for great resource management software? Join forces with Mosaic.

mosaic | The Bigger Picture

mosaicapp.com