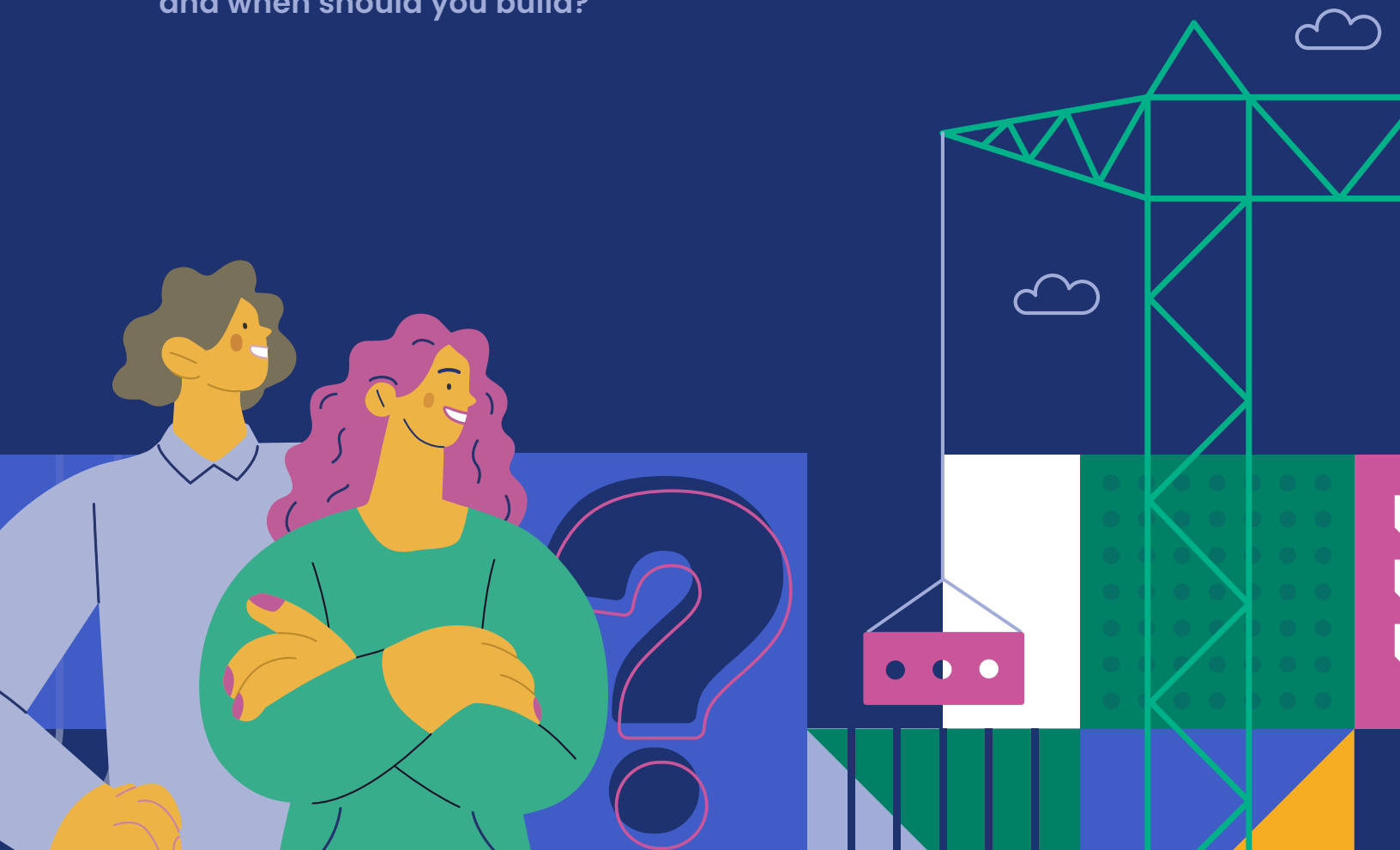


# Build vs. Buy

## Choosing the Best Path for Government Technology

Government leaders are successfully fast-tracking modernization with SaaS solutions. When is it right to buy, and when should you build?



## Choosing the best path ...

**As government leaders chart a course toward modernizing agency operations and digital resident services, many assume they need to build their own software to satisfy their unique requirements and business processes and integrate with legacy systems.**

But the landscape is changing, and tech leaders are adjusting their approach:

Building custom platforms and applications in-house or with a partner was the default approach in public sector organizations for many years, but things have shifted as [software-as-a-service \(SaaS\) solutions have emerged](#) with increased low- and no-code composability and easy API integrations – some purpose-built for government, others adapted for public sector use. It's estimated that by 2025 [over 35% of government legacy apps will be replaced](#) by solutions developed on low-code application platforms.

Along with a need for rapid modernization, leadership is confronted with a range of challenges: constrained budget, evolving community requirements, and ambitious business objectives in the agencies they serve. The public sector is also in the midst of a [talent crunch](#) that can present additional challenges for building and maintaining custom software systems.

**That leaves leadership with a decision: Is it better to build a system in-house or buy a composable software system from a vendor?**

If you zoom out, it seems like a simple choice: If the software solves a problem only you have, build it. If solutions on the market already address your problems, buy them. But it's really a multifaceted decision that depends on the IT team, current tech stack, budget, and the features your community needs.



## Define your goals and needs

### Do you *really* need custom software?

Sure, agency processes differ, but that doesn't necessarily mean you are a unicorn forced to build a solution in-house. Coding a software solution from scratch takes time (sometimes years) before it adds value to your agency's operations. With the advent of composable, cloud-based SaaS, it's more likely than ever that there's a solution already developed that can be configured and leveraged quickly, saving internal IT horsepower for projects that really require it.

Before deciding, the most important thing to clarify is **your desired outcomes and needs**.

#### Your needs might be:

- ✓ Secure payment processing
- ✓ Transaction monitoring and reconciliation
- ✓ AI-driven chat flows
- ✓ Data analysis
- ✓ Improved user experience
- ✓ Identity management
- ✓ Improved security
- ✓ Increased customer satisfaction

#### Some possible outcomes:

- ✓ Reduced support costs
- ✓ Fewer manual processes
- ✓ Faster revenue collection
- ✓ Higher employee satisfaction
- ✓ Fewer support calls and tickets
- ✓ Worry-free compliance
- ✓ More self-serve options for residents

It's important to scrutinize a proposed project's requirements and scope carefully. Consider whether any customizations included will have a meaningful impact on the outcomes the project is meant to achieve.



## Resident tech adoption gets you results

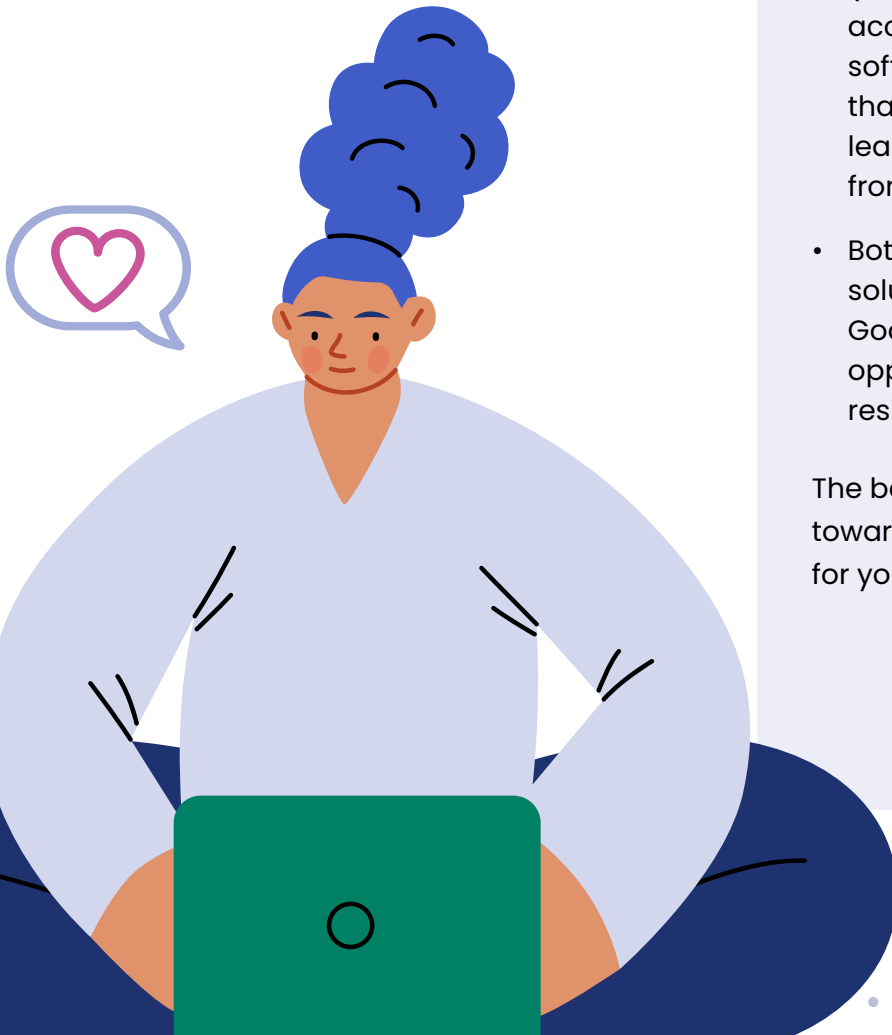
Outlining and documenting your specific needs and measurable outcomes gives the agency clarity, but it doesn't guarantee results. Adoption by residents is the way to maximize the return on your investment. That means that [user experience is critical in driving any outcome](#) (whether that's increased ROI or fewer support tickets).

### Deploying new technology can have a positive impact on your career

While it's not the most pressing issue when choosing between building software and buying off the shelf, it's worth pointing out the effect the process (and final result) can have on the decision maker's career path. Here are a few points to consider:

- Deploying a SaaS solution can be career-altering — in a good way. You'll almost certainly launch sooner, provide a better resident experience, and avoid over-committing your staff.
- Building an in-house solution successfully (on time-ish, close to budget) is a huge accomplishment. But building government software often takes more time and money than expected. It could end up causing leadership and the technical team to suffer from burnout.
- Both building in-house and buying a SaaS solution offer the opportunity to upskill staff. Good leaders take advantage of those opportunities, proving they can take not only residents but also their team into the future.

The best way to build your career is to lean toward the path with the best odds of success for your agency and residents.



## Factor in these key criteria

Evaluating functional requirements and features is a great first step. But other needs such as security and compliance, IT demands, staff skills, and innovation possibilities shouldn't be overlooked — failure to consider these criteria could drain your budget and frustrate staff, or worse yet, leave you without an MVP (minimum viable product). Just as a build requires careful consideration, buying software requires due diligence regarding the product, team, and vendor reputation.



### Speed and innovation

SaaS vendors can provide scalable solutions that add new features and improvements over time. These companies have the team (and financial incentive) to deliver quickly and improve their product continuously. Unless you have a large IT team, building a solution will likely take longer to launch and iterate.



### Usability

Even if your team has the technical skills for a build, it's crucial to consider whether or not user-friendly design is your core competency. A functional, technically sound product won't be of any value to your stakeholders or residents if it's not easy to use. (Remember: Usability ties back to tech adoption, both inside your agency and among residents.)



### On-staff talent

Does your team have the expertise required to build — then maintain and improve — the system? An in-house build will require a deep bench of subject matter experts such as software engineers, product managers, UX designers, fintech or payments experts, integration managers, infosec experts, quality assurance staff, and more. Even larger agencies may have trouble staffing and retaining a team that large, especially given the public sector talent crunch.



### Security

In-house doesn't automatically mean more secure. In fact, the opposite often proves true. With an in-house build, the agency is solely responsible for protecting against malware, data breaches, phishing attacks, and other hacking attempts — which requires constant monitoring and regular testing and patching. On the other hand, solutions that use government cloud solutions benefit from [enhanced security and lower costs](#), minimizing risks for the agency.



### Compliance

Do you want to own the responsibility and cost to stay PCI DSS compliant, or does it make more sense to offload it to a vendor with stringent guidelines already in place? An in-house build would mean that your team is responsible for the cost of the initial assessment and certification, as well as upgrades, annual tests, and recertification.



### Integrations

Most (if not all) SaaS vendors will have good integration capabilities. A good tech vendor will be able to make integrations work. But if you build, you'll have to create an API or work with a developer to explore current government API databases that can fit the bill.



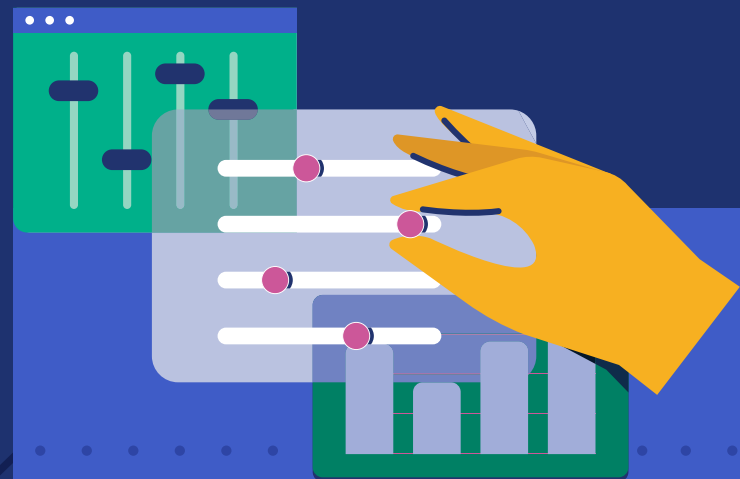
### Customization

For leadership looking to solve a complex and truly unique use case, there is a small chance that no available solution can solve it, and some custom building is required. However, this is increasingly rare. Given the wide availability of composable software platforms and low-code solutions for process automation, a thorough market scan should precede a determination that a custom build is truly required. Although every government agency has unique business rules and processes, these rarely require custom software to satisfy.



### Vendor lock-in

It's natural to worry about getting stuck with the wrong vendor. But it's important to keep in mind that even if you decide to buy a solution from a vendor, you still own your data. That means if you renegotiate terms or decide to switch vendors, you can take it with you. However, always double-check vendor contracts for specifics on the portability of data. Pro tip: never work with a vendor that says you don't own your data.



## The true costs of building

Cost is one of the biggest considerations in a build versus buy decision, and a simple cost estimate is a good place to start. But over time it will be important to consider the total cost of ownership (TCO) of a solution balanced against the outcomes it drives. And factor in a cost avoidance estimate.

Remember: Building a digital customer experience and payments solution from scratch can take months to years. And, unless the scope is incredibly clear, the project could drag out longer than expected.

### Can your budget handle these costs?

**Math to build:** The average salary for a public sector software developer working in the United States is between [\\$120,000.00](#) to [\\$145,000.00](#) per year – and that’s not including the cost of other benefits. Multiply that, since you’ll need more than one engineer. And along with the rest of the hires, you’ll also be paying for hardware.

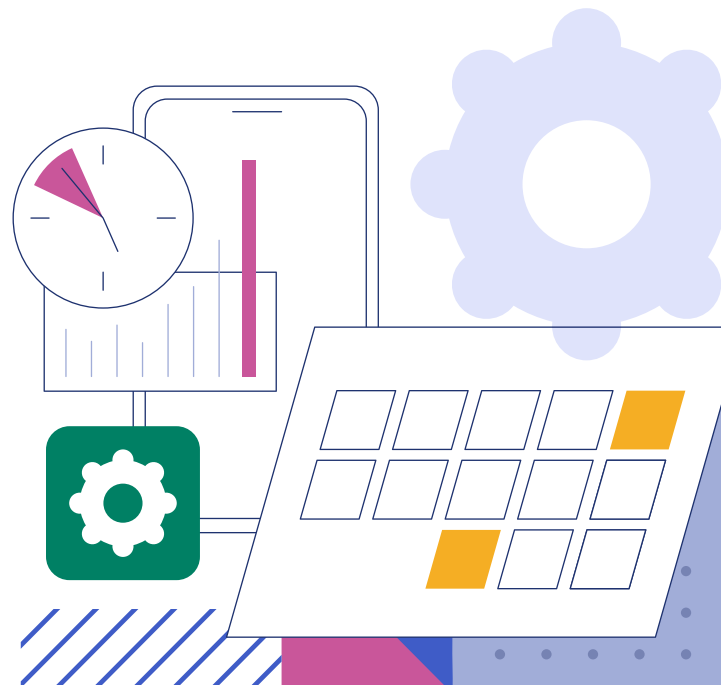
**Time to value:** Let’s say you decide to build a new CRM system to improve customer engagement and help boost revenue. The agency’s initial costs won’t pay off until it sees improved customer relationships and increased revenue. Can the department’s budgets absorb the development costs until a viable project is complete? (For reference, [public-sector projects take 3.9 years to complete](#), on average. Larger, more complex projects could take longer.)

**Maintenance:** The costs don’t stop after a system is launched. Building software requires continued investments to upgrade, add, or change features; address tech debt; and maintain compliance certifications.

### Opportunity and strategic business cost:

Will an in-house build keep employees from solving other core business problems? Will waiting months or years for a custom-built solution decrease financial and business opportunities? A build can go awry in many ways, using up staff’s valuable time and [potentially wasting millions of dollars](#).

Non-completion is also more of a risk with custom-built solutions – [a large percentage fail](#). And in-house builds almost always cost more upfront, are expensive to maintain, and quickly become outdated. (Unfortunately, that means taxpayers pay more and internal processes are hindered).



## Moving forward: What's right for your agency?

Consumers are accustomed to having most other services (health care, banking, entertainment, etc.) at their fingertips, and they expect intuitive digital government service delivery. Deploying tech solutions can be a “move fast or fall behind” game. In some instances, it might be better to partner with a specialized vendor that can amplify your team and speed up your goals.

There will be engineering work regardless of whether your team chooses to build or buy, but the two paths require very different levels of time and expertise. A fully custom build could be the right answer, but it will come with a higher cost and a lengthy timeline, and not every agency has the time, staff, or budget to handle an entire build. Composable SaaS technology is a more efficient, scalable, and flexible solution, especially for smaller IT teams or agencies wanting to quickly modernize the user experience.

Given the rapid pace of tech innovation and AI improvements, it's more important than ever to consider your modernization projects carefully. And when you couple the speed of technology changes with a continuing talent crunch, your decision gets a little more tricky. Whether you buy or build, ensure your chosen path can keep up.

### Additional resources:

[READ NOW](#)[READ NOW](#)[READ NOW](#)



## Build vs. Buy: Questions to ask yourself

---

### Features

- What features are needed now? What about in the future?
- Are priorities clear? Have we decided on “must have” vs. “nice to have” vs “don’t need” features?
- Are future innovations in the picture? How will those be handled?

### Technical

- What does our current tech stack look like?
- What’s our data source(s)? Are they accessible?
- Are we capable of integration if needed?
- Are there suitable solutions already available on the market?

### Speed of delivery

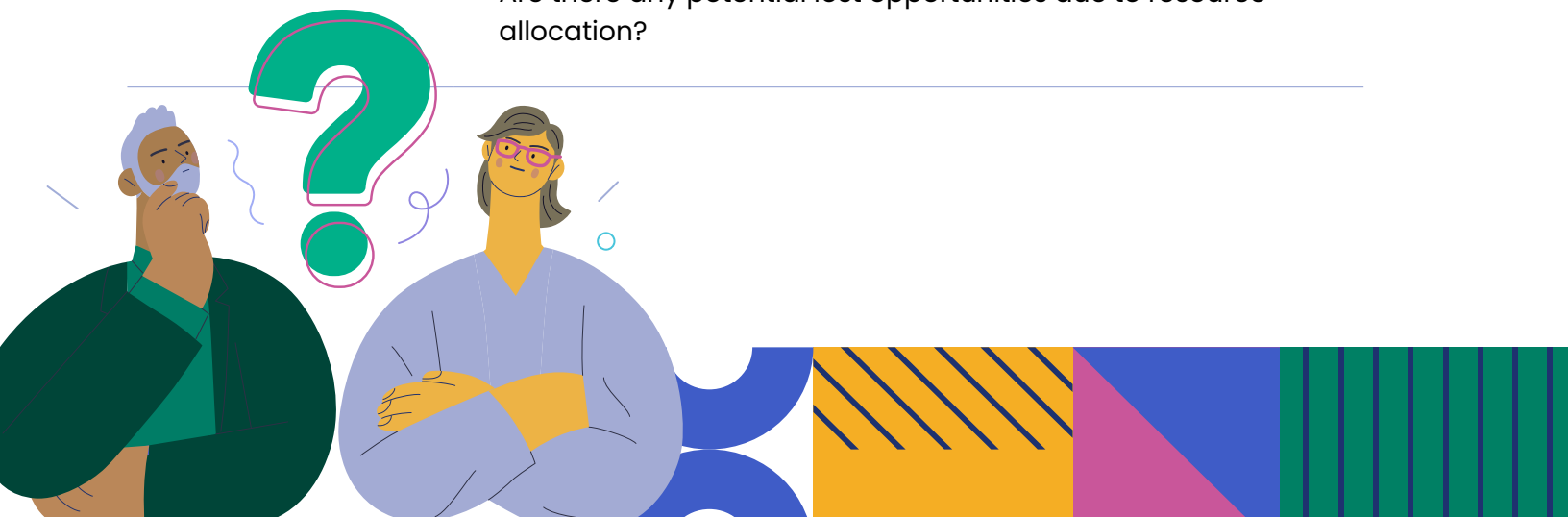
- What is the timeline?
- What’s driving the timeline?
- Can the team deliver a solution on time?
- Are there consequences of delivering sooner vs. later?

### Design

- What are our user needs (internal and consumer)?
- Do we have the knowledge to prioritize user experience?

### Cost

- Have we accounted for **all** costs?
- How much does speed to delivery accelerate ROI?
- Are there any potential lost opportunities due to resource allocation?



## About PayIt

PayIt enables state and local government agencies to deliver a great resident payments experience that accelerates the shift to digital. Agencies choose PayIt to better achieve their mission through improved operational efficiency, customer support, and resident satisfaction. Our solutions span property tax, courts, utilities, DMV, outdoors, and more. PayIt provides a single resident profile across agencies and jurisdictions, integrates into back-office and adjacent systems, and our team helps clients drive adoption of digital channels. Serving more than 100 million residents in North America, we have received awards from Fast Company and StateScoop, and have been listed in the GovTech 100 for 8 years and counting.



Smarter for government.  
Easier for everyone.

To learn more, visit [www.payitgov.com](http://www.payitgov.com)

