Top 10 Build vs. Buy Objections and How to Overcome Them

This document lays out common objections that might arise related to the Build Vs. Buy argument and how to handle them. Don't outright crush objections – instead, debunk misconceptions and provide insights by sharing information, offering different perspectives, and clarifying the value of Okta vs. a homegrown solution. We recommend that you engage Presales support as necessary when going through Build vs. Buy total cost of ownership (TCO) conversations with customers and prospects. Presales can help address any technical questions or objections that might arise and encourage multi-faceted discussions.

Table of Contents

General Objections

1. We can build a simpler and cheaper version ourselves.

Start by introducing the common pitfalls of a DIY approach and follow up with a Build vs. Buy TCO comparison.

Common pitfalls of building in-house:

- Underestimating the complexity of a complete identity service
- Oversimplifying requirements – scope creep happens as use cases expand and users demand greater functionality/features
- Keeping up with evolving standards, requirements and technologies related to identity
- Tying down key developer resources on building and maintaining a service that is not core to the business
- Finding/hiring expert identity and security resources, of which there is a significant shortage in the market
- Ensuring enterprise-grade security requires ongoing dedicated expertise

Has your prospect considered all the elements that go into the project? Very often, key areas are overlooked which can lead to cost underestimation. Use the TCO tool to shed light on all the components behind a customer identity service and demonstrate how a DIY solution is often the costlier option.

2. We have dedicated development resources for our project so this is an already sunk cost. / Or / 3. If all vendor solutions require some coding, why not just build it all ourselves?

The way companies deploy their developers is critical to their success. Developers working on the right things can make companies excel in their respective markets. But for this to happen, they need to focus on building functionality and features related to the core business / application – not identity.

Additionally, companies often underestimate the specialized technical expertise required for these projects and how there is a critical shortage of that expertise in the market. Help prospects investigate the following: how experienced and up-to-date is their team on security and DevSecOps? How do they plan to stay on top of the latest cryptographic trends? How will they find and hire experienced security professionals?

Lastly, companies must consider the ongoing efforts required to maintain, update, scale, and innovate on a homegrown solution. Use the TCO tool to support your arguments with cost estimates related to both the build and maintain phases.

4. We can rely on open source solutions for software development. They're easy to use and free.

Open source solutions are deemed to be "free" because they have no upfront license costs. But there are many hidden costs and risks that can add up quickly and that organizations should be taking into account, including:

- Developer time to install the software, try it out, test it, find any bugs, and evaluate security and vulnerability of the components

- Ongoing DIY maintenance and future-proofing as there's no support service built-in to these solutions
- Exposure to security risks related to the lack of automatic updates and patching of vulnerabilities found in open source code.

Tool-Specific Objections

5. We own our TCO and don't need Okta to model it.

Our tool is simple and easy to use and requires minimal effort from prospects. We've already done all the heavy lifting by capturing all of the components and associated costs that go into building a comprehensive customer identity solution.

At a minimum, we can provide a polished report with a TCO analysis based on average industry costs – with no strings attached. Prospects can use this to validate their internal numbers, and perhaps even identify components they may have overlooked. Alternatively, we can schedule a brief call to walk through our tool and generate a tailored analysis and report in less than 30 min.

6. We have simple identity requirements; we don't need something sophisticated. / Or / 7. We don't need Social Auth. We don't need authorization. And so forth.

Ask your prospect: "These are your business requirements today, but what will they be tomorrow?" Companies find it hard to predict future use cases or user volumes as their applications take off and, as a result, can become victims of their own success.

Scope creep is a common phenomenon – according to PMI's 2018 Pulse report, over 50% of companies experience uncontrolled changes to their project's scope, up from 43% in 2013. Scope creep itself is creeping up as end users demand more features, more functionality, and more seamless experiences.

In the best-case scenario, fulfilling new requirements will be burdensome for your developers. In the worst-case scenario, it can seriously impact your budget and schedule. And if you get identity wrong, the consequences are not limited to costs – they can extend to reputational risks and customer abandonment.

When creating a TCO analysis, ask the customer to consider including not only the basic requirements they've identified today, but also future needs to address expanded use cases, as well as any advanced features that their end users might demand.

8. Where do these calculations and assumptions come from?

Our model was built in close collaboration with our Product, Security, Finance, and Engineering teams, stakeholders that have extensive experience in the development of a comprehensive identity service. In addition, we've conducted dozens of TCO assessments with prospects / customers, vetting our proprietary inputs along the way with customer identity and access management subject matter experts.

Keep in mind, our model makes the following assumptions:

- Deployment using open source components / software on a public cloud provider such as AWS
- Dedicated development resources for the build / deployment phase of the project (we assume this due to the complexity, scope and scale of identity projects that typically require project management and diverse technical knowledge, including cryptography, database security, system engineering, security auditing, etc.)
- Developers with minimum 5 years of experience including identity (we assume this due to the inherent complexity of identity).

9. Does it really take so many months to build/maintain that?

At first glance, many identity components such as sign-in pages and registration workflows appear simple. However, when building out these components, organizations quickly realize there's a lot more under the hood than meets the eye.

For the most costly and/or time-consuming components within the TCO, please refer to the guidance below. Specifically, pose the below questions to your customer to help them uncover areas requiring considerably more development effort than they may have initially believed.

Security Monitoring and Engineering Practices

- How experienced and up-to-date is your team on security and DevSecOps?
- What are your processes for ensuring that your code and libraries are constantly monitored, maintained, and patched?
- How will you stay on top of the latest cryptographic trends?
- Which organizational controls and secure engineering practices exist in your engineering teams?
- Where do network and application security responsibilities live within your organization?
- Do you open up your code for penetration testing?
- What tools do you use for security analytics?
  ➔ How many FTEs will you need to support the above on an ongoing basis?

Future Proofing and Technical Debt

- How will your team keep up with evolving identity standards and protocols?
- How will you rapidly build and implement additional identity features and functionality as your business needs evolve?
- Who will keep up with changing API back-ends for your application and system integrations?
- Who will be responsible for expanding your integration ecosystem as your business and partner collaborations grow?
- What kind of technical debt have you accumulated over time as your developers raced to meet deadlines?
- How will you manage productivity losses due to key developer turnover?
- Examples of future proofing / technical debt activities: you've discovered a SAML vulnerability in the existing code and will need to patch it; MFA becomes a common compliance requirement in your industry and you will need to go build it; etc.
  ➔ How many FTEs will you need to support the above on an ongoing basis? For all other components, refer to the TCO tool for in-depth descriptions and information on the tasks and efforts required.

10. Okta will also require time to deploy and configure. How are you accounting for this in the tool/calculations?

If this question comes up, mention that we can include the costs of Okta into our tool to show a TCO comparison between Okta and an in-house approach. However, Okta's customer identity products range from no-code (out of the box, easy, and fast to set up and configure) to pro-code (lengthier deployments) depending on the customer's use cases, environment, and preferences. As a result, it is difficult to provide a standard number for the resources and time required to configure and deploy our solutions – these numbers will vary widely from customer to customer. Generally, the following parameters can influence the amount of time/resources required:

- Number of applications supported
- Complexity and scope of identity requirements
- Sophistication / expertise of in-house development team
- Preference for pre-packaged vs. fully customizable identity components
- Proximity of target go-live date

Depending on the use case and how the customer stands up against the above parameters, this is an area where you can benefit from soliciting SE support to come up with an approximate estimation. The tool can then be updated accordingly in the 'Total Cost of Okta' section (e.g. 0.5 FTEs for the initial period, etc.).