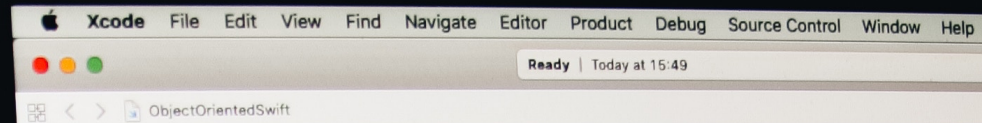




APPTIMIZE



```
init(x: Int, y: Int) {  
    self.x = x  
    self.y = y  
}  
  
/// Returns the surrounding points in range of  
/// the current one  
func points(inRange range: Int = 1) -> [Point] {  
    19 var results = [Point]()  
    20  
    21 let lowerBoundOfXRange = x - range  
    22 let upperBoundOfXRange = x + range  
    23  
    24 let lowerBoundOfYRange = y - range  
    25 let upperBoundOfYRange = y + range  
    26  
    27 for xCoordinate in lowerBoundOfXRange...upperBoundOfXRange {  
    28     for yCoordinate in lowerBoundOfYRange...upperBoundOfYRange {  
    29         let coordinatePoint = Point(x: xCoordinate, y: yCoordinate)  
    30         results.append(coordinatePoint)  
    31     }  
    32 }  
    33  
    34 let coordinatePoint = Point(x: 2, y: 4)  
    35  
    40 coordinatePoint.x  
    41 coordinatePoint.points()  
    42 }
```

What You Need to Know When Deciding to Build or Buy Your Experimentation Software

Table of Contents

[Introduction](#)

[Resources](#)

[Accessibility](#)

[Cost](#)

[Innovation](#)

[Speed to start experimenting](#)

[Final Thoughts](#)

Introduction

You've already come to the conclusion that experimentation is the best way to create the optimal user experience. You know that a better user experience can improve engagement, retention and revenue, and thus helps you grow your product and organizational KPIs faster.

Experimentation testing gives you cold, hard data to answer any kind of question about how to improve your product or service. By presenting your customers with different versions of your product, you can measure the cause-and-effect relationship between any changes and the impact on KPIs.

While A/B testing is simple in concept, it is hard to execute well, especially given the fragmented customer journey across digital and physical channels today. Now that you've made the right decision that experimentation is a priority for your business, you need to choose a framework that will enable you to rapidly run experiments and accurately analyze them.

Should you build or should you buy?

Here are the key considerations that you should keep in mind while answering this question.

How many extra engineers do you have?

It's no coincidence that the very best internal testing solutions exist at the largest tech companies. Netflix, Google, Facebook, and Airbnb have no shortage of star power at their disposal. These companies, which generally take a "let's build everything ourselves" approach, have all devoted significant engineering resources to creating custom A/B testing solutions that match the exact needs of their internal stakeholders and systems.

However, most companies don't have these kinds of engineering resources at their disposal. Making even small changes to your product is likely a headache (more on this later). Carving out an all-star team to build a perfect experimentation solution is probably difficult at this stage, and would require shifting away from other core priorities.

Your team is really good at building whatever product they are building (probably better than our engineers at that!). The basis of all SaaS products is letting the experts build what they build best. In a world as complicated as product development, we believe it's important to have a system built by a team of experts (who have scaled a product for the likes of Viber, Vevo, Strava, and Wall Street Journal).

Who at the organization needs to be able to use it?

One big problem with building in-house software is that it's often not "democratic," or accessible by the people who need to use it to achieve their KPIs. There are multiple teams that have an ongoing interest in using experimentation to meet their goals (product, marketing, engineering, and data to name a few), and the usage of the tool will vary amongst these different stakeholders. Satisfying the requirements of different users can be very difficult, something that requires a deep understanding of each of these personas and their goals.

For example, without building in an intuitive, plain-English results dashboard, if someone on your Marketing team wants to analyze the impact of an A/B test, they may need to understand in-depth statistics to evaluate the results well. Many home grown A/B testing solutions don't have a dashboard to display results, so someone in analytics has to write SQL to pull data, posing a barrier between the end user and impactful data.

Without a system optimized for efficiency, product managers will be limited in their efficiency and velocity — whether it's launching an experiment or interpreting the results. Given that the results of an experiment have meaningful impacts on business KPIs, this technical constraint can be a serious roadblock in your growth.

If all you have is a hammer, everything looks like a nail. And if you're a software engineer, every organizational challenge is an opportunity to write some custom code. This approach has its benefits, as engineers can build functionality using the exact specs and terms that the organization is used to.

Building your own solution might seem like the right thing to do to have a high level of control on the product, but in the long term it can slow your team down, make you less agile, and increase organizational dependencies on engineering.

What is the cost of ongoing maintenance?

The cost of building your own platform isn't just limited to the upfront development; you also have to consider the ongoing maintenance. [Studies show](#) that 75% of the total cost of software can be attributed to maintenance. Around a quarter of that goes to fixing bugs once the software is deployed, another quarter goes to adapting the code to a changing business environment, and the remaining half goes to enhancing the feature set for future iterations. You incur all of those future costs when you ship a

"light-weight" custom A/B testing framework, because the implicit expectation is that you will fix, adapt and evolve the framework as the business grows.

Beyond the considerations listed above, organizations also need to consider the cost of infrastructure (included in the costs of investing in a SaaS system), ongoing testing / QA, and scaling adaptations as the product grows.

How fast do you want to innovate?

Your rate of learning and innovation is determined by how quickly you can experiment. If every experiment you run requires an engineer to write some custom code, you will likely run fewer experiments.

If you do have the time and resources to build a successful experimentation framework, it's important to think high-level and long-term about the requirements that you will have down road as you develop a robust culture of experimentation. We have found that in-house solutions are often built for a specific use case and are ultimately limiting as the experimentation roadmap develops. Without years of knowledge built up through practice and observation, it's hard to come up with an end product that will fit nearly all possible edge cases.

How fast do you want to start experimenting?

You are likely working with a robust product roadmap, one that requires you to be agile and iterate quickly. Since the ultimate goal is to improve your product and experimentation is the key to doing this effectively, the goal should be to minimize the time that it takes to get your experimentation program up and running. The sooner you start experimenting, the sooner you can get results, and the sooner you can make improvements on your product.

When buying an existing solution, integration and implementation is seamless and typically takes less than a few weeks. Building an experimentation solution, however, can take months-- even years. And while you're building in-house, your competitors are already experimenting, iterating and growing.

Final Thoughts

When considering if you should build or buy software it's important to consider the time and resources that you have at your disposal. Oftentimes, companies choose to test the waters by utilizing an SDK solution so that they can begin testing right away and avoid any opportunity cost.

Third-party A/B testing softwares, like Apptimize, have many advantages that you should consider before building in-house: greater accessibility for everyone at your company, lower costs over the platform's lifetime, and a team with in-depth experience in testing and mobile development. By purchasing a modern experimentation solution like Apptimize, companies are able to save time and money. Most importantly, you are able to focus your internal resources on what they do best-- building an awesome product.

Benefits of Buying	Why Companies Choose Apptimize
Developed by engineers who are experts in and focused on experimentation software	Apptimize is reliable and can scale with you as your user base grows. With nine patents issued and pending, and over 50 person-years of development, companies like Wall Street Journal, Viber, and Hotels.com come to Apptimize for a experimentation solution that really works.
Use a tool that is accessible to all types of users	Make quick changes with our codeless visual editor or with our dynamic variables to save you time. Often the simplest tests can be the most impactful. Non-technical users can launch experiments involving copy, CTAs, images, navigation items, and more in a matter of minutes through our

	<p>patented, most robust WYSIWYG Visual Editor in the market. Dynamic Variable and Code Block experiments allow teams to test anything they can code.</p>
Experiment with a tool that's constantly innovating and adapting	<p>Built for Mobile Product Teams. Apptimize has built our solution based on feedback from hundreds of customers across the globe and various industries. This gives us unique insight into what product leaders are testing and what's working. We benefit from this collective wisdom to come up with features, experiments and ideas that our customers use to improve product experience.</p>
A cost-effective solution	<p>Best-in-class mobile support. Apptimize offers a complete experimentation program solution, including visual testing, programmatic testing, results and analytics, and release management. Along with these experimentation essentials, you will receive support from our expert digital strategists, who will help you get started, analyze results, and mature your experimentation roadmap.</p>
A robust experimentation program that accommodates to your needs	<p>Easy and intuitive interface allows you to test quickly and efficiently. More testing leads to more growth. Apptimize is designed to help the whole product team test more efficiently.</p> <p>The results? Customers who run more than eight experiments per year with Apptimize grow over two times faster than the industry average</p>
Ability to start experimenting quickly	<p>Apptimize fits in seamlessly with your full development stack. Apptimize automatically pulls in events from most major push and analytics providers so your team can get going even faster.</p>

Apptimize empowers product teams to efficiently run A/B tests, roll out and manage new features, and deliver personalized user experiences. Based in San Francisco, Apptimize is backed by US Venture Partners, Google Ventures, Y Combinator, and others.

Join companies like Vevo and Wall Street Journal; fuel data-driven growth for your mobile app with the best-in-class mobile A/B testing. [Request a customized demo today!](#)