ionic

Hybrid vs. Native

An introduction to cross-platform hybrid development for architects and app development leaders

Contents

> Demand for mobile experiences is growing 5x faster than internal IT teams can deliver.
>
> Gartner

The Mobile Delivery Gap

Back in the early days of mobile, there was really only one way to give users the performance and features they expected. You had to use a native SDK.

Of course, that came with tradeoffs:

- Building in parallel for each mobile platform
- Managing multiple codebases
- Hiring and retaining highly specialized and costly native developers

Meanwhile, demand for mobile experiences has grown exponentially. By 2022, 70% of all enterprise software interactions are expected to occur on mobile devices.[1]

Given the time and cost of traditional native development, it's no surprise that many app dev teams are struggling to keep up.

Thankfully, times have changed. As mobile and web technology have evolved, hybrid has emerged as a viable alternative to native. Many are now looking at hybrid development as a way to simplify and speed up development. Let's take a look.

[1] Market Guide for Mobile Application Testing Services - Gartner, June 2017

The Choice to Go Hybrid

The growing adoption of hybrid is evident in a recent Forrester survey that found two-thirds of mobile developers are choosing a web-based approach over native tools.[2] Meanwhile, top brands like MarketWatch and Nationwide have chosen hybrid over native for their flagship consumer apps.

The top reasons for making the switch from native to hybrid, based on independent research and testimonials, are:

1. **Speed.** Building for multiple platforms from a single codebase often makes delivering cross-platform apps 2-3x faster than native.
2. **Efficiency.** Forrester estimates that hybrid can save an organization between 75-80% in support and porting costs compared to native.
3. **Omnichannel.** Hybrid apps can run anywhere the web runs - on a desktop or mobile browser, as a mobile app, or PWA.

Put together, these advantages have helped centralized app dev teams close the gap and better satisfy the demand for mobile apps for customers and internal employees.

> Forrester estimates a hybrid approach will save an organization between 75–80% compared to native.

[2] Native, Web, And Cross-Platform Mobile Apps All Have Their Place - April 5th 2016 Forrester

What is a Hybrid App?

Hybrid apps are essentially native apps. They're downloaded from a platform's app store or marketplace, and access the same native features and hardware-based performance acceleration as any app built with a native SDK.

The key difference is that hybrid apps are built using open web technologies like JavaScript, HTML, and CSS, rather than the proprietary or specialized languages used by iOS, Android, and others. That means anyone with a web developer skill-set can build an app using the hybrid approach.

Hybrid apps run in a full-screen browser, called a webview, that is invisible to the user. Through customizable native plugins, they can access the native features of specific mobile devices (such as the camera or touch ID), without the core code being tied to that device.

That means hybrid-built apps can run on any platform or device, all from a single codebase, while still delivering a native look and feel.

Comparing Hybrid vs. Native

In the sections that follow, we'll provide a one-to-one comparison of hybrid vs. native, highlighting the pros and cons of each approach.

It's important to keep in mind that the decision to choose hybrid or native should be based on the unique goals of your organization, the circumstances of a given project, and composition of your existing development team.

Of course, we're betting big on the power of the web and the promise and potential of cross-platform hybrid development. But we understand that individual circumstances should drive which route you choose.

Hopefully this ebook will give you some helpful tips to help you choose the right approach for your next project. But keep in mind, there's often plenty of room to accommodate different approaches in a single organization. It's not always an either or decision.

Why Hybrid?

Write once,
run anywhere

Use the talent you
already have

Deliver a great
user experience
across platforms

Build for
the future

Let's take a look at each of these.

Write Once, Run Anywhere

Rarely is a mobile app only designed for a single platform. Consumers, partners, and employees all have a choice of platforms and devices.

Following the native approach, that meant you needed to build separate apps for each mobile platform, and sometimes specific apps for tablets and smartphones.

This is where hybrid development shines.

With a hybrid framework like Ionic, you can run your app on any platform or device, all from a single codebase. Ionic also provides platform continuity, so the look and feel of your app isn't one-size-fits-all. It automatically adapts to each platform. And because hybrid technology is web-based, you can run your app in a regular browser as a Progressive Web App, or PWA.

That means your users get a great experience across platforms and devices, and you only have one codebase to worry about.

> **"**
>
> **Until recently, it was quite common to program Android apps in Java and iOS apps in Swift/Objective-C.**
>
> **Multiplatform frameworks get rid of having to maintain two code bases, and they are thus becoming very popular.**
>
> InfoWorld - October 2017
> Apple's Swift is losing developers to multiplatform frameworks

"

According to the 2017 Stack Overflow Survey, only 6.5% of all developers cited Swift and Objective-C as familiar languages. In contrast, web developers made up 72.6% of respondents, and JavaScript appeared as the most commonly used programming language in the survey.

Ionic Blog - July 2017
In-House Teams & Talent Rising to the Challenge

Use the Talent You Already Have

The web developer community is about 30x greater in size than the number of native mobile app developers. Many development teams already have a deep bench of programmers who understand HTML, CSS and JavaScript.

Why not leverage the talent you already have in-house to build your next mobile apps?

With a hybrid framework like Ionic, your existing web teams can build high-performance apps that run on any platform or device, using the tools and technology they already know and love.

That's a lot easier than recruiting, training, and hiring specialists. Plus, centralizing on a single skill set makes it much easier to reassign teams when a project is finished - whether that's a desktop web app or another mobile project.

The Best UX Across Platforms

Before Dow Jones MarketWatch made the switch to hybrid, their app store reviews were in bad shape. Now, they're consistently around 4.5 out of 5.

Native advocates will claim that only a native approach can deliver the speed and performance that you need to create a great UX. Sure, we think that's important as well, and hybrid-built apps offer the same hardware-based performance acceleration as native apps.

But what the MarketWatch team found is that user experience isn't just about performance. Simplifying app dev and consolidating onto a single codebase means more time to add features, fewer defects, and more time to fix bugs that find their way through.

Most importantly, a great UX means a consistent app experience as your users move between platforms, devices, and modes of interaction, including mobile and desktop browsers. Only hybrid can deliver that kind of consistency.

> **When we were working natively our user satisfaction scores were like 3 stars on Android. All time low was 2.5 on iOS. Now we are at 4.5 stars on iOS and near 4.5 stars on Android.**
>
> Brian Aguilar,
> MarketWatch - May 2017
> On switching from native to hybrid with Ionic

Moving now to the web platform offers you richer, more innovative options moving forward.

Building for the Future

Development organizations are tasked to build applications for the future.

Future applications will be run on a growing number and diversity of platforms oriented around the web: wearables, IoT devices and other M2M communications like GPS in cars, asset tracking systems, or portable medical devices.

Thankfully, the web is the most widely used application runtime in the world.

Hybrid development taps into this mainstream. Using a forward-looking development stack lays the groundwork for you to seize opportunities beyond today's devices, like Progressive Web Apps.

Moving now to the web platform offers you richer, more innovative options moving forward. How you capitalize on these opportunities will be the measure of your success.

Of course, hybrid applications are not without their drawbacks:

1. System overhead

The use of the webview may introduce a degree of overhead compared to native. The abundance of performance APIs and increasingly powerful hardware have made this less of a factor in recent years, but it's still something to consider. For most applications, the difference in performance is hardly noticeable. But for 3D games and other performance-intensive applications, hybrid may not be the best choice.

2. Third-party plugins

Hybrid apps are able to access nearly every native feature of a device, like the camera or gyroscope, by using native plugins. Open source Cordova plugins are the most popular. The use of these plugins does add complexity to development. Ionic offers a library of the most common plugins, known as Ionic Native, that are easier to use than standard Cordova plugins. Nonetheless, this is a factor to consider.

3. Framework dependencies

Choosing a cross-platform approach means you're placing trust in the framework vendor to keep up with the latest and greatest native features and design patterns of each mobile platform. While Ionic is committed to keeping up with new Android and iOS versions, there's still a dependency.

Why Native?

Performance

Rich native library

No third-party dependencies

Let’s explore the common reasons for choosing native.

Benefits of Native Development

As stated previously, native is still the preferred approach for many mobile developers. And there are some good reasons for that. While part of that is based on the legacy of having few viable alternatives, native still has its advantages today.

Performance
Native code is still faster than Javascript and HTML. This matters when developers are looking to build demanding graphical applications such as games and other intensive animation applications. Mobile browsers are coming closer to bridging the gap for these types of intensive applications using WebGL specification; however, native still has the advantage here.

Rich native library
Using native SDKs allows the developer to access the latest features specifically designed for those platforms, without the complexity of dealing with native plugins. This is key when you need to provide a rich contextual user experience such as facial recognition for iOS or touch ID for Android.

No third-party dependencies
By building exclusively with a native toolset, developers aren't bound to any third-party to keep up with support, and there's not as much of a dependency on open source communities like Cordova to keep up with the latest features.

Here are the most frequently cited challenges associated with native development:

1. Longer development cycles

Native apps usually have a longer development cycle, especially when building for multiple platforms, which requires two or three different code sources for iOS, Android or Windows. Each platform has its own nuances which require specific changes, updates and maintenance which bloat the cost of an application and add development time. This creates a lot of iterations within the development process in order to customize and test for each platform, it also reduces your agility to launch your application or push updates.

2. High development costs

If you haven't caught it before, developing mobile applications natively is expensive and time-consuming, mostly driven by the time it takes to build for each platform, along with the cost of hiring and retaining talent.

3. Native talent hard to find

Finding and hiring iOS and Android developers is difficult, time consuming and expensive and it's also difficult to repurpose those developers for other projects outside of mobile.

Native apps usually have a longer development cycle, especially when building for multiple platforms.

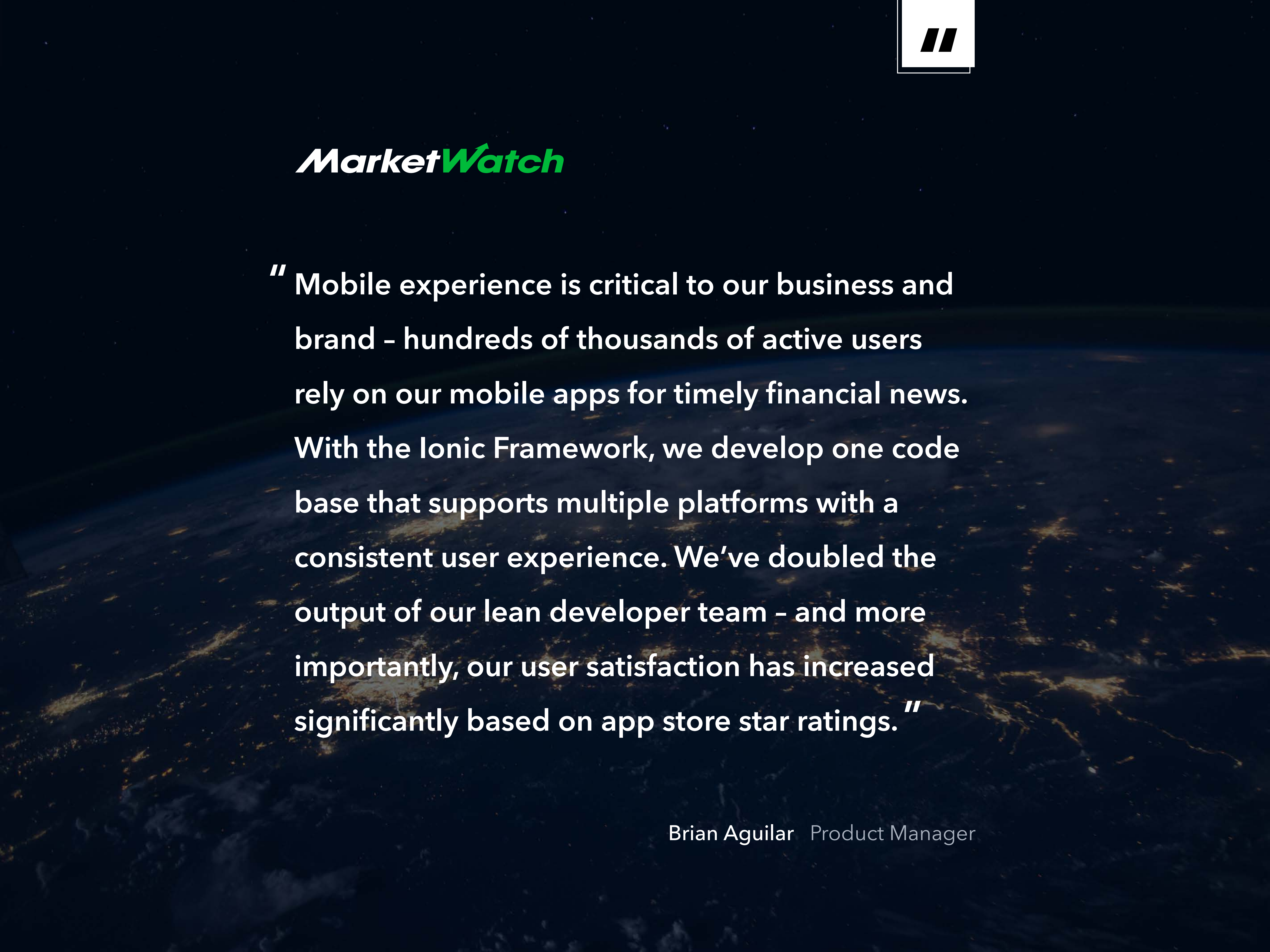
Attribute	Native Approach	Cross Platform Hybrid
Developer Skill Set Needed	Objective-C, iOS SDK, Java, Android SDK	HTML, CSS, Javascript
Distribution Method	App Stores	App Stores Desktop Browser Desktop App (e.g. Electron) Mobile Browser Progressive Web App
Speed to Develop	Slow	Fast
Development Cost	High	Low
Maintenance Cost	High	Moderate/low
Graphical Performance	High	Moderate
App Performance	High	Driven by use case
Access to native functionality	Full native library	Full native library (requires third-party plugins)
UX consistency across platforms and devices	Requires separate apps	Yes

A Little About Ionic...

With Ionic's open source Framework and powerful suite of developer tools and services, you can run your app on any platform or device, all from a single codebase. Ionic also provides platform continuity, so the look and feel of your app isn't one-size-fits-all. It automatically adapts to each platform. And because our technology is all web-based, you can even run your app in a desktop browser as a progressive web app, or PWA. That means your users get a great experience across platforms and devices, and you only have one codebase to worry about.

With over **4 million apps** built on Ionic so far and **5 million developers** using the platform, Ionic is the platform of choice for any organization looking to develop beautiful apps that provide a rich user experience with minimal investment.

But don't just take our word for it...



MarketWatch

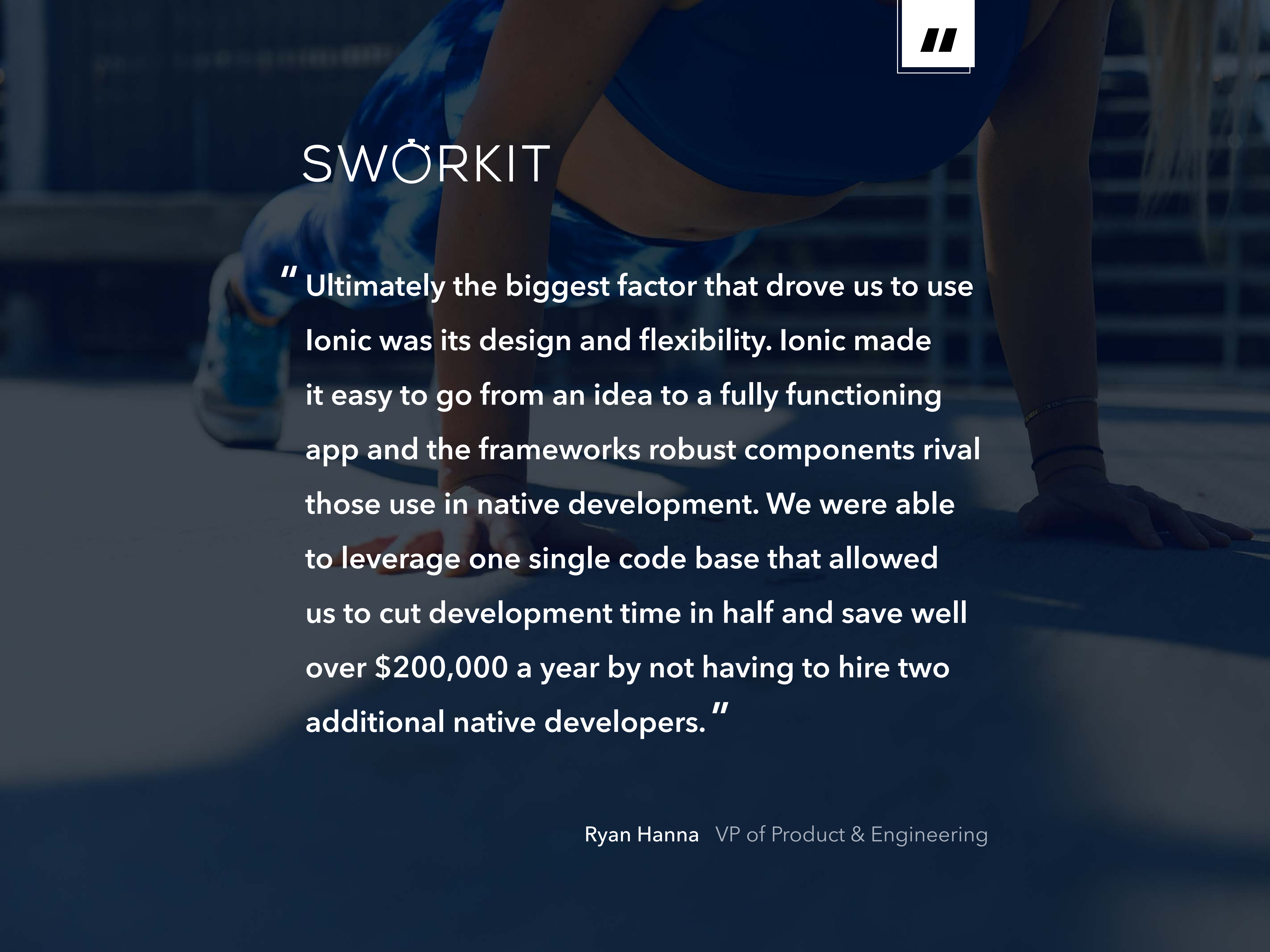
"Mobile experience is critical to our business and brand – hundreds of thousands of active users rely on our mobile apps for timely financial news. With the Ionic Framework, we develop one code base that supports multiple platforms with a consistent user experience. We've doubled the output of our lean developer team – and more importantly, our user satisfaction has increased significantly based on app store star ratings."

Brian Aguilar Product Manager

AIRBUS
HELICOPTERS

> “ The Ionic Framework and its cross platform approach to mobile app development is very efficient and effective for us. With Ionic Creator, we are also able to collaboratively design with our business stakeholders. Being able to share working designs and prototypes with the business in the design phase improved our delivery time by 2X. ”

Heinz Haeussler Mobile Technical Expert



SWORKIT

> “Ultimately the biggest factor that drove us to use Ionic was its design and flexibility. Ionic made it easy to go from an idea to a fully functioning app and the frameworks robust components rival those use in native development. We were able to leverage one single code base that allowed us to cut development time in half and save well over $200,000 a year by not having to hire two additional native developers.”

Ryan Hanna VP of Product & Engineering

Let Us Help With Your Hybrid Strategy

Ionic makes it easy to build high performance apps that look and feel beautiful on any platform or device. The Ionic Framework is the #1 adopted cross-platform development framework in the world, with a vibrant community of more than 5 million developers in over 200 countries. Connect with one of our App Strategists to see how we can help.

Let's Talk Strategy