INSPIRING APPS

A Business Perspective on Building Mobile Apps

SECOND EDITION

INSPIRING APPS

A Business Perspective on Building Mobile Apps

Foreword to the Second Edition

It is hard to believe that it has only been three years since we wrote the first edition. As cliché as “time flies” may be, it sure applies to the mobile industry. Since the first edition, mobile hardware has improved, with higher resolution displays, faster processors, more interconnectivity, and smaller, more personal wearable devices. At the same time, software has advanced at a dizzying rate to keep up with the pace of the hardware enhancements. It is an exciting time to be a mobile app developer.

As amazing as these technology changes are, it is equally exciting to us to think about how mobile technology may help improve your business— or how you may even build an entire business around a mobile app. Neither is a small task, and I hope that you will find guidance and inspiration in this publication to help you navigate the mobile app waters.

While we cover the latest mobile technology, perhaps the most significant addition to this second edition relates to marketing. People refer to the early “gold rush” days of the App Store, during which time there were noteworthy success stories about independent developers and small companies who got rich quickly, not only because they built a good app, but because the market was starved for them and demand far exceeded supply with only thousands of apps in the App Store.

Today, the App Store and Google Play tell different tales. There are still success stories— some of them quite spectacular. But the app you release will be among about 1.5 million of them in both the App Store and Google Play. That means that you can’t expect to simply release your masterpiece and wait for the deposits to roll in. In fact, your journey to app success will start with the marketing effort you invest long before a line of code is written. That is the subject of the greatly expanded chapters 2 and 7.

Even with the incredible numbers of apps in the stores for each platform, we know there are apps yet to be written. Our hope remains the same, that through this book we might help you sift through the terms and options, providing some insights to get your project started. Should you have questions or want to discuss an idea, please contact us at info@InspiringApps.com.

- Brad Weber, President and CEO of InspiringApps

Foreword

In the early 90s, I remember a professor who loved to thumb her nose at people who were so presumptuous as to talk on their car phone while driving. She kept a plastic PlaySkool phone in her car which she used to mock passersby. My, how far we've come! Not only is it commonplace to see mobile phones everywhere, but their capabilities are greater than the personal computers of the PlaySkool phone days. Making phone calls may be what you spend the least amount of time doing on these new phones.

On January 9, 2007, Apple changed the smartphone market forever when Steve Jobs announced the iPhone. It was a revolutionary change which has left an indelible mark on the industry. Hundreds of millions of iOS devices later, and with the addition of Android, BlackBerry devices and Windows Phone 7, you control your world with a touch.

With the wild proliferation of apps in the stores for each platform, it may be hard to imagine there are apps yet to be written. But new apps continue to be released, and the pace doesn't seem to slow. And the stores don't provide any visibility into the enterprise app development phenomenon that's also growing in leaps and bounds.

Businesses around the globe build apps for their own internal use that are distributed privately to their team, without passing through the public stores. That's why we imagine you are reading this book. You have an idea about an app that hasn't yet been created—one which will impact your personal life or your business. And you need help sifting through the dizzying array of terms and options to identify the best way to either bring your app to market or deploy it to your team. We hope this book will get you started down that path.

For everyone at InspiringApps, this effort is a matter of good stewardship. Working with clients entering the app world for the first time, we have learned a lot about the common questions decision-makers have on their minds. We are pleased to share what we have learned in hopes that it will help you enter this exciting arena with greater confidence. We have prepared a second edition of this book because so much has changed since we published the first edition. There are new questions and new options to consider. We've done our best to help you make the most of them.

- Brad Weber, President and CEO of InspiringApps

CHAPTER 1

APP BASICS

This chapter provides a basic overview of terms, definitions, and concepts that are foundational for app development projects.

APP BASICS

In This Chapter

App Definition

"App" is short for "application." It is a self-contained program or piece of software designed to fulfill a particular purpose. The term gained popularity in reference to software modules created for use on mobile devices, first for smartphones and subsequently for tablets. After its rise in popularity on mobile platforms, the same term is now also used to refer to software that runs on a PC or Mac computer. When we use the term "app" in this book, though, we are referring solely to software for mobile devices.

Many apps are developed for the purpose of making some task, or group of tasks, easier and more streamlined. Others are created solely for good old-fashioned fun, like games and social interaction apps. Some apps are built for the masses while others are custom developed for a particular person or organization. This distinction ties into a question we often hear— "Do I need a custom app?"

Off-the-Shelf vs. Custom Apps

In order to decide if you can use an "off-the-shelf" app or if you need custom development, it's helpful to

understand the benefits of each. An off-the-shelf app is one with broad usefulness, produced to address the needs of multiple people or organizations with similar challenges. It is often developed by a third-party who sells the technology via one of the app stores.

Depending on your needs, budget, and strategy, something off-the-shelf may address your requirements nicely. In fact, many off-the-shelf business apps are designed to allow a moderate amount of customization, enabling you to align field names, dashboards, and other basic functionality with your preferences or those of your company.

Some off-the-shelf products are acquired through a one-time purchase, and others are sold on a subscription basis where users pay a monthly or annual fee to use the app. The latter is sometimes called software-as-a-service (SaaS) and SaaS products exist for mobile apps in the same way they do for desktop and web-based applications. For example, there are a number of SaaS apps that provide all-in-one packages to assist small- and medium-sized businesses in capturing leads, tracking lead profitability, and marketing.

If your requirements are complex and specific to your business, you likely need to go the route of a custom-made app. A custom app is one built specifically for your organization to meet a well-defined set of goals. Frequently an outside developer or development team is employed for the creation of a custom app. The benefit of a custom app is that it does exactly what you need, in the way you need it.

It’s worth noting that such custom solutions don’t have to be large and complex. Apps that integrate and exchange data with other company tools can prove beneficial in filling the gaps between existing systems. For example, many companies use Salesforce for customer relationship management (CRM) and have smaller complementary apps built to either extend the system, or present a more limited set of tools, focused on the specific needs of their workforce.

The bottom line is, if you can find an existing product that meets your needs, use it. If your needs are unique, and unmet by current app offerings, then it is time to explore developing a custom app to satisfy your business requirements.

Devices, Operating Systems, and Hardware

While most people don't give much thought to what's happening behind the scenes on their mobile device, it's important for you to understand mobile hardware and software components associated with phones and tablets so you can effectively determine how your app will fit. Variability and fragmentation in these components not only present challenges to developers, but to you as well, as you consider the platforms you want to support.

Devices

The device is the physical item you hold in the your hand. The iPhone took the mobile market by storm. In the early days, it was the only game in town. There was one set of device capabilities. One screen dimension. One operating system (more on operating systems in a moment). It took time for competing device manufacturers to release their own hardware, and for companies like Google and Microsoft to release compatible mobile operating systems. Now that others have joined the race, there are seemingly limitless options from which to choose.

When Apple released its iPad, they added a second "form factor" to target in addition to smartphones: tablets. Now there are many more choices, including smaller tablets like the Barnes and Noble Nook and Amazon Kindle Fire HD, as well as larger smart phones like the Samsung Galaxy and HTC One. As you think about developing an app, you will need to decide which form factors you desire to support. Consider the users' environment when they complete the task at hand:

- Will they need to operate the app with one hand, suggesting that a smaller device may be more appropriate?
- Will they need to interact with, view, or make decisions based on more information than can fit comfortably on a smartphone screen at one time?
- Can your content and features be tailored to fit a wide variety of form factors?
- Will your app serve a variety of users with different needs, giving you valid reason to support a wide variety of form factors?

Operating systems

There are currently three prominent operating systems (OS) for mobile devices: Apple's iOS which runs iPhones, iPads and iPod touches; Google's Android OS which runs on a wide variety of smartphone and tablet devices; and Microsoft's Windows Phone 7/8. Of the three, iOS and Android currently lead the pack by a considerable margin

over Microsoft in terms of market share.

Native apps are developed in programming languages native to these operating systems, meaning the app will only run on devices for which it is specifically written. If your target audience predominantly uses a particular device, say an iPad, you can develop for only that platform. If you want your app to run on multiple devices using multiple operating systems, multiple versions of your app may be required.

Web technologies and cross-platform frameworks make it possible to more easily develop an app that will run on multiple operating systems. Learn more in the Cross-Platform Development section in Chapter 3.

If you are building an app for internal use at your company, your operating system options may be dictated by your corporate IT group. If you are developing for the market, you will need to take into account the adoption numbers for your target users and develop your app for the dominant operating systems. Global percentages may not apply to your target demographic.

Hardware

Depending on your idea, you may find you need the device to have additional hardware capabilities. Is a device camera required? Some devices have a camera while others don't—that is an easy one to identify. For more complex capabilities, you may need to rely on your development partner to help determine if your app requires an accelerometer to detect motion, a gyroscope to detect orientation, or a GPS to detect global position.

Mobile Apps vs. Mobile Websites

As you enter into the world of mobile, you may wonder what the difference is between a mobile app and a mobile website. Adobe answers this best in their Inspire newsletter: "A **mobile application** is a software application that works on a specific mobile device's operating system and is downloaded to the device to perform a specific set of functions. A **mobile-optimized website** is a site that is intended to be viewed using a mobile browser (e.g. Safari, Chrome, Internet Explorer) on the various display sizes of phones, tablets and other mobile devices."

The following table provides a good contrast between mobile websites and mobile apps.

	Mobile Website	Mobile App
Accessed on handheld device (smartphone or tablet)	✓	✓
Downloaded and installed on mobile device	✓	✓
Runs on any device with a browser and internet connection	✓	
Takes advantage of device-specific capabilities and functions	✓*	✓
Full interactive capabilities		✓
May run without an internet connection	✓**	✓
Development tools and languages	HTML, CSS, JavaScript	Java, Swift, Objective-C, JavaScript, C#, and more

* Certain device-specific capabilities available depending on development technologies utilized. For instance, mobile websites can take advantage of location services, leveraging the device's geo coordinates with the user's permission.

** HTML 5 supports offline use in mobile and desktop browsers, but users must visit the site at least once first before the offline resources are available for the user's browser to cache.

In other words, mobile websites often display information from your main website in a way that is mobile-friendly, meaning when someone accesses your website from their mobile device they are presented with a version of your site which is visually optimized for their smaller mobile screen size. Content is often pared down from the main site to what is essential for mobile visitors.

In contrast, while some apps are created to imitate site functionality in a completely controlled environment (e.g Amazon's shopping app or the Facebook app), a mobile app may not have anything to do with your website. Companies often create apps to do things like boost brand awareness or drive sales. The Starbucks app is a good example of this, as it enables users to earn rewards while buying product at their stores.

As you consider the basics of app development, you will also want to consider your mobile strategy. Developing an

app should take place in the context of a broader mobile strategy— which itself should should roll-up to align with your marketing and business strategies. If you haven't yet thought much about mobile strategy, consider this article for some background.

CHAPTER 2

MARKET CONSIDERATIONS

The explosion of app publishing has been quite stunning over the past few years. The commercial that proclaimed, “There’s an app for that” is increasingly accurate. This chapter will introduce important market and business analysis work to complete before you embark down the app development path.

MARKET CONSIDERATIONS

In This Chapter

Understand Your Market

Define the opportunity

Before launching into an app development project, it's critical to take the time to clearly define the opportunity you have in mind, and reflect it against what solutions already exist. This is necessary regardless of whether your target user is an internal team or a consumer external to your company. It's even true for gaming apps, as differentiation is always valuable.

Opportunities exist when there is a gap between what is currently on the market and the possibilities that economic changes, technology advances, or new social trends open up. Developing a product to fill that gap is part science and part art, but it always involves creating a solution that is perceived as useful, usable, and desirable. While that may sound externally oriented, even apps aimed toward improving internal productivity or systems must meet this criteria. If your employees don't perceive a benefit from using your app, they won't bother.

The more you can do at the outset to define the opportunity you see, the better success you will have at actually creating the right product and knowing how to

market it. This process, sometimes called customer development, is a way to reduce business risks by challenging assumptions about who the customers are, what they need, and why and how they do their work.

In the book, *Lean Customer Development - Build Products Your Customers Will Buy*, Cindy Alvarez notes that “customer development is critical to success but grossly underutilized.” The book offers a practical education in customer development, noting that only customers make a product successful. As Alvarez states, “without customers willing to buy a product, it doesn’t matter how good or innovative or beautiful or reasonably-priced a product is: it will fail.”

If not already involved, it’s important to bring your marketing team into the process at this point. They can make a critical contribution to the planning and development right from the start by helping you to gain this outside-in perspective on the concept you have in mind.

Through various types of analysis, marketers can generate accurate and relevant information about consumer preferences and target markets. This research can then be translated into product specifications that have the greatest likelihood of connecting with consumers. Every hour spent defining the customer pays dividends in the development lifecycle by enabling you to know which features are critical and which are not. Further, your marketing team can help you understand market cycles, branding, and other industry dynamics so that your company can make smarter decisions on everything from launch timing to product price.

To help define the opportunity, your team will want to work together to consider questions like these:

- What problem will this app solve? (your value proposition)
- For whom will you be solving that problem? (your target market)
- How big is the opportunity? (your market size)
- How do they address the problem currently? (minimum expectations)
- What features are critical for success? (requirements)
- Why are you best suited to do this? (your differentiator)

Determine if a comparable app already exists

Once you have defined the problem you want to address

as clearly as possible, you need to learn about the solutions that are currently available. If you desire to sell an app externally, these alternatives will become your competition, and you will need to quantify what unique value your app will bring relative to them. If you are looking for a solution to gain internal efficiencies, this research will tell you if you can use an off-the-shelf app, which could save you money and/or time, or have something built that is tailored to your unique needs.

Thankfully, the digital age has made basic market research much easier to do directly, without the need for subscriptions to specialized market research databases. While you still may choose to buy a targeted report at some point, you often can get a preliminary feel on your own.

One of the best places to begin is simply on your favorite search engine. Look for products using relevant keywords that describe the functionality of interest. You will also want to directly search Apple's App Store and Google Play in a similar fashion. Barnes and Noble and Amazon also have their own app stores that are worth exploring.

If you find apps that seem relevant, download them on the appropriate device and experiment with them so you can see their strengths and weaknesses. Reading reviews will also give you insight into potential gaps in functionality that might be opportunities for you— or red flags if you were planning to use the app in your organization.

If a similar app does already exist, you have to consider whether developing something new is worthwhile. In the case of an externally-facing app, your product, at a minimum, will need to offer capabilities that are distinguished from the competition. Even better, though, would be to offer features so unique that they essentially create a new category for a particular niche of the market. Otherwise you will find yourself in a price war in order to gain market share, and this is a place from which it is hard to truly "win." In the case of an app for your organization, your custom app should offer quantifiable gains in efficiency or provide other critical functionality over the existing options, in order to justify the investment.

Know your industry

While understanding the competitive landscape can give solid insight into obvious hazards opposing your market

entry, there is no guarantee of success if current competition is weak. Markets are constantly changing and buyers, suppliers, and other companies are all dynamic factors influencing who will prevail and profit. As a result, it's important to have a robust understanding of the market for any apps you desire to sell.

The details of such analysis are beyond the scope of this book, but if the investment cost is high for developing and marketing your app, we encourage you to learn before you spend. Understanding things like industry structure, trends, growth rates, barriers to entry, etc. are all essential to ensuring your app is a successful part of achieving your company's business goals.

Consider Your Business Strategy

Ensure your app aligns with business priorities

Let's say that your study of the external landscape leads you to believe there is a viable opportunity at hand. Before charging down the path, make sure you are able to make a solid case as to why this app development project is the right thing for your company. Business strategies set priorities within companies, and the success of your app development project depends upon how well it helps your company to achieve its objectives.

Some questions to ask yourself include:

- What business objectives do you hope to meet through this product?
- How does it fit into the portfolio of products you already offer?
- How does it complement your brand?
- How does it serve your primary customers?
- How does it leverage your expertise?
- How does it improve current processes or systems? (*for internally-oriented apps*)

Whether your app is internally or externally oriented, it must fit into the larger vision of your company in order to garner appropriate resourcing and attention. Likewise, aligning your ideas with these priorities will only help to sharpen your focus and increase your likelihood of success.

Set goals that define success

Once you are sure your app project will advance your company's priorities at a high level, you need to more specifically define what it will look like to succeed* with your new app. (*We are speaking of business, not technical, success at this point. Your market and

customer segment research should identify features that you believe the app needs to offer, and we will further unpack some of those technical considerations in the next chapter.)

Goal setting is a complex process, and a robust business plan is not needed right from the start. There are too many unknowns and you need to work within that ambiguity, not pretend you have it defined. Nonetheless, thought should be given to how you are going to approach the process. Many companies break down broad business objectives into smaller strategic initiatives that are achieved through individual goals and tactics. Regardless of how your company chooses to do it, core goals for you app must be tied to one of the business objectives, and must be realistic to achieve within a reasonable time frame. Trying to do too many things only dilutes your efforts.

Once you know the goals, it's important to determine what critical performance variables you can evaluate to measure your success. Clear objectives enable your sales and marketing teams to know what matters the most and thus make decisions to support reaching those goals.

In summary, as you consider the market for your app idea, you are trying to ascertain three things: (1) is the opportunity real? (2) can your company win at it? and (3) is it worth it? Obtaining answers to these questions will not only provide you with valuable business insights, but also enable you to well define your specifications as you enter into development.

CASE STUDY

We have had the privilege of working with many great client partners on many great projects. The technology is interesting. And our partners seem to have good ideas that will solve real, challenging problems in the market. However, far too often, those partners don't achieve the market success they are striving for.

We want to contribute to the success of our clients. That is one of the primary reasons we placed so much emphasis on marketing in this edition of the book. We think solid marketing practices present a great opportunity for improvement to many app development projects. It takes a lot of effort to avoid these common pitfalls:

- I already know my customers, and I know what they will want in an app. We'll get their feedback when the product ships.
- I'm not sure we have enough features yet to make this app compelling. Let's add more before we ship.
- We can't start design or development yet because we don't think we have yet defined the exhaustive list of requirements for the first release.

To counteract those detractors, we strongly encourage you to get real customers involved in your process as early as possible, and to release to them as frequently as you possibly can so they can provide feedback throughout the life of your project. Doing so will allow you to make course corrections before your course is too difficult to change to meet market demands.

If you haven't already, read *The Lean Startup* by Eric Ries. Eric does a terrific job of cataloging common mistakes of startups and enterprises alike as they fall short in their pursuit of market success.

CHAPTER 3

TECHNICAL CONSIDERATIONS

There are numerous operating systems, development frameworks and tools. This chapter will provide a high level overview of the technical considerations involved in an app development project. Understanding the basics will enable you to make informed decisions about how best to apply technology to address your business challenges.

Chapter 3

TECHNICAL CONSIDERATIONS

In This Chapter

Development Tools, Technologies, and Frameworks

Operating systems

One of the first decisions you will make from a technology perspective is to decide which mobile operating system to use for your app. This is necessary because iOS apps and Android apps are not written using the same programming languages. iOS apps are written in Swift and Objective-C while Android apps are written in Java.

You might consider market share for each platform to inform your decision. Android has captured a majority of the total mobile OS market share. Apple's iOS is second with Windows, BlackBerry, and others trailing well behind. If you intend to produce an app for the general public, and you don't know anything about their mobile operating system preferences, let those overall market numbers be your guide. However, if you are serving a population that you know has a bias that is different than the global totals, follow the habits of your target market.

If comparing the total size of the installed base doesn't lead to a clear choice, consider operating system upgrade adoption rates. Apple has a remarkable track record of getting its users to upgrade to current versions of its operating system. At the time of this writing, about 6 months after its release, 77% of iOS users are running iOS 8, which is the latest version of Apple's mobile operating system. Another 20% are running iOS 7. That results in an astounding 97% of users on the two most current versions of iOS. This adoption rate is a great benefit to developers because they can focus on current technologies, tools, and capabilities, without having to worry much about backward compatibility.

By contrast, Android adoption is much slower. On that platform, users don't always get their operating system upgrades from Google. Distribution of updates may be governed by a mobile carrier or device manufacturer. About 4 months after its release, Android 5.0 (Lollipop) has just over 3% of the platform's share. Android 4.4 (KitKat) has 41%. So combined, the two most recent versions of the Android operating system represent just 44% of that market. In fact, if you include the six most current operating system releases from Google, you still don't quite reach Apple's 97% adoption rate.

Development tools

Operating system vendors including Apple, Google, Microsoft, and RIM, all know that a thriving app

ecosystem on their platform is gold. Customers rarely buy a mobile device for the default apps that ship with it like email, web browser, photo management, etc. While those apps are important, people are more drawn to the apps that will aid them with a work task, enrich their personal life, or both. Hopefully your app will fit into one of those categories.

In order to build your mobile masterpiece, you need tools. And each of the leading operating system vendors offers tools to do just that. A precedent was set, when desktop development was thriving, to charge license fees in the thousands of dollars for the tools required to develop apps on Mac and Windows operating systems. Those days are long gone. Now the operating system vendors see more value in having a huge number of apps on their platform than they do in making money from license fees charged to developers. For that reason, the development tools from those vendors are typically free. But don't think free means that the tools lack in quality. In fact, they are quite excellent, and the vendors continue to make significant investments to improve them and make it easier to build powerful solutions on their platform.

Developer tools often include one or more desktop applications that allow designers to lay out app interfaces, and developers to write, compile, debug, and test code. Those applications are integrated, designed to work well together, and provide everything you need to take an app from concept to submission to an app store or distribution internally in your enterprise.

Operating system vendors do charge a nominal fee to submit apps to their app stores. For example, Apple currently charges $99 annually per company, regardless of the number of developers on the team or the number of apps they submit to the App Store. Fees for other vendors are comparable.

Cross-platform development tools

You may need to develop an app that runs on more than one operating system. If you develop apps using the native language of each platform, you'll need to developer or team with expertise in each of those languages. Alternatively, you can lean on one of the many third parties that has aimed to simplify cross-platform development. They build a framework around a single programming language (e.g. JavaScript, C#, Ruby, and more) and provide a single set of tools from which developers can create apps to run on multiple of the most

popular operating systems.

There is a lot of interest in the cross-platform tool space among investors. Consequently, there are a lot of cross-platform tools, all vying to win in a potentially lucrative market. Currently, some of the top players include:

- Native cross-platform app development
  - Xamarin C#
  - Appcelerator / Titanium JavaScript
  - Kony JavaScript
- Game development and immersive experiences
  - Corona Lua
  - Unity C#, UnityScript (2D & 3D)
- Mobile web development
  - Sencha JavaScript
  - PhoneGap JavaScript

Part of the appeal of cross-platform tools is their focus on a limited subset of complete platform features. They make it relatively easy to get started. And if your requirements perfectly overlap with what they do well, you can save time. Some developers may be intimidated by the vast frameworks of native development tools, so they take comfort in the limited and focused capabilities of cross-platform alternatives.

The downsides of cross-platform tools

On the surface, cross-platform tools sound like a big win. Why would I write something twice or three times, when I can just write it once? However, there are trade-offs and pitfalls to consider in cross-platform development.

Cross-platform tools offer, in part, the promise of time savings from writing once and then running on multiple platforms, without writing code again from scratch. Unfortunately, a significant amount of platform-specific development and testing is still required, especially if your app's needs don't perfectly overlap with the capabilities of the framework, even though the core development language is shared across the platforms. You may still realize time-saving benefits from cross-platform tools, but that is more likely to come from not having to hire or retrain your developers on a new language than it is to come from efficiencies in the tools themselves. For example, if you have a team of web developers who are already very familiar with JavaScript from their web work, they may be delighted to contribute to your mobile app development efforts using a cross-platform tool that uses JavaScript as its language of choice.

Further, cross-platform tools are usually not free. It is very expensive to build and maintain tools that allow developers to write apps for multiple platforms with the same source code. The vendors have to recover their costs, and can't do so through device sales or percentages of app sales like the native platform tool vendors can. Instead, many do so by charging developers a fee for the tools, support services, or back-end services like analytics and crash reporting for deployed apps.

Operating system vendors have their own roadmaps for their technology. They don't consult with the third party cross-platform vendors as they improve their offerings. So those third-party vendors are always playing catch-up. They hear about new operating system features at the same time the rest of the public does, but then they have to add support for those features in their products and test extensively before releasing them to their own developer communities. Given that they have limited resources, cross-platform tool vendors identify the features they think will be most popular and only add support for those.

Depending on the vendor, the gap between their platform's features and those of the core operating system can grow quite large. Developers can close that gap by developing modules to supplement the features of the third-party framework. Those modules are likely to be written in the programming language native to the platform (e.g. Swift, Objective-C, or Java). That means that your development team will need to have skills in the native languages to get the most from the cross-platform tools. If you find yourself in that situation, it grows more challenging to prove the cost or development time savings from leveraging cross-platform tools.

It is one thing to develop a framework with support for the language features developers need to build compelling apps. It is an entirely different level of effort to develop the tools that allow designers and developers to visually lay out the interface for their app. In some cases, cross-platform tool vendors don't offer visual interface layout tools at all. In other cases, they are available for an extra charge. Visual interface design tools provide instant feedback and are similar to the graphic design tools, like Photoshop and InDesign, designers are already accustomed to. By contrast, without a visual layout tool, designers and developers are likely to have to describe the app's interface in a series of text-based instructions which are far more time-consuming and error-prone. The

lack of a visual layout editor can erase all of the time saved developing with cross-platform tools.

Deciding between options

At a high level, we covered two main options in the previous section:

1. Develop your app using platform-specific languages and tools, keeping in mind that doing so requires a distinct development effort for each platform you hope to serve.
2. Develop your app using cross-platform frameworks and tools which enable your single app to function on a variety of operating systems.

Regardless of your choice, find a well-rounded development partner who excels in more than one technology. People and process more directly impact the success of a project than the technology. Poor developers can make a mess of the most sophisticated, elegant tools. And exceptional developers may find ways to make magic with less shiny tools. We recommend a well-rounded partner because of the old adage, “If you have only a hammer, you see every problem as a nail.” If your partner has multiple tools in the toolbox, they are more likely to recommend a solution which is a good match for your challenge and not just a good match for their limited technology options.

Your development partner will work with you through the discovery process (see Chapter 5 for more on discovery) to understand all of the options and make the best determination for how you should move forward in development.

CASE STUDY

Many years before founding InspiringApps, Brad Weber was asked to advise on a desktop development project that was struggling. Although the products referenced in this study are quite dated, the message still rings true today.

The development had built a product using FoxPro. Customers complained. It was difficult to use and unstable, crashing frequently. The team sought advice about rewriting the product in 4th Dimension (4D), which they believed was superior to FoxPro. Brad shared his positive experience with 4D, including the quality solutions that had been built for customers on the platform. So the team embarked on a lengthy and

expensive effort to rewrite their product, only to find themselves in a similar spot. The new product was also difficult to use and unstable.

Had Brad steered them wrong? Was the technology garbage? No. The problem was not the technology, but rather the team. Plenty of teams had delivered great desktop applications with both FoxPro and 4D. However, unskilled developers will make a mess of a project, no matter how sophisticated the tools. The valuable lesson to take from the experience is that your choice of partner is probably far more important than any particular tool in their toolbox.

Brad later learned that the same team went on to rewrite their product once again— in Java. Although he never learned whether their third attempt was a success, using the same team would be unlikely to lead to better results.

The Cloud

The term “cloud” as it pertains to computing is 10 to 15 years old. Today, “the Cloud” typically refers to remote servers used for computing and storage. Instead of storing data on your company’s dedicated internal server and networking infrastructure, using the cloud means storing that data remotely. In other words, from a more personal vantage point, rather than storing data on your computer or other device, your files are stored on servers outside your home. These servers are accessed via the internet, which means you can reach your data from any device if you have internet access. Programs like Dropbox and Evernote are cloud-based services commonly used by consumers. Many mobile apps require some use of cloud infrastructure, either to limit the amount of storage used on somewhat limited mobile devices, offload intensive computations to more capable servers, or to enable sharing of information among users.

Cloud resources are often shared by multiple entities, but they don’t have to be. A “Private Cloud” is a collection of hardware (still remote) that is dedicated to a single group, company, or organization. The “Public Cloud” is comprised of virtual private servers. Although there are physical machines providing computing and storage services, those services are partitioned so they may be shared by many tenants of the same physical hardware. Each tenant installs their own virtual server, resulting in an environment that feels to each customer like they are running their own private server in the cloud, when in fact

they are likely one of many customers partitioned on the same hardware.

Cloud providers like Amazon, IBM, and Rackspace take responsibility for managing all of the physical assets. If hardware fails, they repair or replace it. If network connectivity suffers, they restore it. For many companies — even those building apps— it is a tremendous relief not to have to tend to server hardware. Hardware failures still happen but minimal service disruption occurs because cloud providers quickly move a failed virtual server to a different physical server and the system is back up and running.

General purpose cloud services allow you to install your favorite server operating system and run any kind of service you'd like on it. However, there are special purpose cloud services as well. iCloud, from Apple, is a good example. iCloud provides computation and storage services to developers and end users of Apple devices. You can choose to backup your mobile device to iCloud. You can store your photos in iCloud so they sync automatically across devices. You can store documents in iCloud for sharing across devices as well. But you can't install your own software to run on iCloud servers. You can't allocate more CPU horsepower to your iCloud account. Apple has defined all of the services that are available via iCloud and you can't add your own. Although iCloud provides services to Apple devices (iMacs, MacBooks, MacBook Pros, iPhones, iPod touches, and iPads, etc.), Apple announced at their 2015 developer conference (WWDC) that they were providing an API to allow developers to provide access to users on other platforms to take advantage of their iCloud services.

Cloud security

Security is one of the reasons some IT departments may still prefer to manage their own servers. We don't think cloud servers are inherently less secure than physical hardware under your own control. But because cloud servers are publicly accessible, you will still need to be diligent about protecting the resources you store there— whether that is application source code, database data, or other sensitive files. Limit access to the servers as much as possible. Require authentication to access sensitive resources. And then try to minimize the damage that can be done if an outsider was able to authenticate by using data encryption, access logging, and more.

AppSync

InspiringApps leverages the cloud for most of our projects. We find that many customers want their app to work offline, sharing data with others seamlessly behind the scenes when a network connection is available. To facilitate offline use, and that kind of distributed content management, we created our AppSync framework. It includes a database (or many) in the cloud that represents the “truth”, or the master collection of data shared by all users. Learn more about AppSync on the InspiringApps website. <link here to AppSync page on the new IA site>

Security

Depending on the nature of your app, you may need to secure data— for financial transactions, medical patients, and corporate secrets.

Encryption

Just because you can’t see the files on your mobile device as easily as you can on your computer or USB drive doesn’t mean that they aren’t accessible to thieves. If your app will store any data or files that are at all sensitive, encrypt them. Encryption is the best protection against a thief with a stolen device.

HIPAA

If you store any personally identifiable health-related data in your app, you are likely subject to HIPAA regulations. We can’t cover all of the details in this book, but you should be prepared to encrypt sensitive data, limit and log access to it, and be able to notify affected individuals if data is compromised (e.g. on a lost or stolen mobile device).

PCI - Payment Card Industry

If you process credit or debit card transactions or store credit card data, you are subject to PCI compliance. In-app purchases on a mobile device are often required to use approved platform tools to avoid credit card processing. For example, Apple and Google both provide in-app purchase frameworks to allow developers to charge for features, subscriptions, and more. Instead of entering credit card information to complete a transaction, users provide a user ID and password for the platform. Apple and Google batch purchases for a single user and charge a credit card that the customer has on file with them, avoiding the need for you to touch credit card information or worry about PCI compliance.

LDAP integration
LDAP (Lightweight Directory Access Protocol) is one of many directory services a company might employ to manage user accounts centrally. You might need to add support for directory services for an enterprise app, allowing users to authenticate in your app with the same credentials they use to access other company resources. It is a common task, so you are likely to find the resources you need through internet searching.

Integration with Third Parties

Depending on the intended use of your app, you may not have to develop all of the app features yourself. Rather than create your own file management system, you may choose to integrate with a service like Box or Dropbox. Rather than build full-featured accounting capabilities into your app, you may want to integrate with Quickbooks. And if your company is already managing all of its sales contacts, campaigns, and touches in Salesforce, you can leverage the data that is stored in that environment without replicating it.

Companies offering services like those mentioned above are eager to have third-party developers integrate with their tools, extending their platform and increasing your shared customer's investment in it. To that end, they frequently publish mobile SDKs, web service APIs, and associated documentation to make the process easier for you and your team. An API (application programming interface) is a specification that describes the way in which apps can connect to the device and exchange data with it. They may also create a library of code, or framework, that provides all of the plumbing to handle the connection and data exchange. This is known as a Software Development Kit (SDK). In that case, developers add the vendor's SDK to their app project and typically provide the user interface to take advantage of the features.

Connecting with Others in an App

There are many ways to exchange data between mobile devices. All require a network, but the network may take many forms. The internet is one of those networks. Two devices can either connect directly to one another via the internet or pass data through some kind of proxy, like a server (see the Cloud section earlier in this chapter). Email and SMS fall into the latter category. You don't connect your mobile device directly to the recipient's device to send an email message. Instead, you connect to a server that routes your message to the recipient's mailbox. At

some later point, the recipient's device retrieves that message. Our AppSync data synchronization framework also falls into the latter category. Users submit locally added and changed data to a server where it is later retrieved by others on different devices.

Devices on the same local wifi network can discover one another, connect, and exchange data. Some multi-player, multi-device games take advantage of this capability. If you want to race your friend head-to-head in your grand prix game, you might invite her after your app discovers that she is also running the same app on the same wifi network. The data exchange will include your position on the course, speed, score, etc.

Mobile devices can also be connected via Bluetooth. Traditionally, that process involves pairing, which could take 15 to 30 seconds. If you have paired your mobile device to a headset, vehicle, keyboard, or something similar, the process is the same.

One emerging connection method that we are particularly enthusiastic about is mobile mesh networks. Apple calls their implementation Multipeer Connectivity. Google calls theirs WiFi Direct. Both provide support for mobile devices to create an ad hoc network using a combination of Bluetooth and wifi where no network exists. Imagine you are camping with friends who are equipped with similar mobile devices. The devices can create their own mesh network, allowing the campers to play their favorite networked game, exchange photos, create a shared camp responsibility checklist, and more. Network traffic will be confined to the group of campers. A mesh network doesn't magically provide internet access where none was previously available. It does, however, provide the ability to exchange data with nearby devices relatively easily where there was no network at all.

Unfortunately, at the time of this writing, Apple and Google only want you to camp with friends who have the same mobile operating system. Currently, Multipeer Connectivity and WiFi Direct are not compatible with one another, so you can't easily exchange data across a mix of iOS and Android devices. It is possible to do so. It is just a lot more work than relying on the convenient frameworks provided by Apple and Google. FireChat is a cross-platform chat solution that chose to write their own sharing code.

What about bumping? Some apps and operating systems

support data exchange by tapping two mobile devices together. The bits don't fly out of your device directly into your friend's device when a bump happens. Instead, apps that provide such a service typically send the app event along with your geolocation and precise time to a server. That server makes a determination, based on the location and time, of the devices that most likely bumped one another. At that point, the server can act as a proxy to facilitate the data exchange. If there are lots of users in a relatively small space, all bumping around the same time, it will be difficult for the server to make the correct determination. Imagine participants at a large conference all being encouraged to exchange contact information during a break in the schedule. The result may not be what is intended.

App Performance

Crash detection

Crashes happen. Ideally, you witness and correct all of the issues during your development process, but apps may crash in the field after they are released. You will want to know about those crashes so your team can quickly address them and ship updates to your users. There are a remarkable number of crash detection and reporting tools, including Crittercism, Crashlytics, Bugsense, TestFlight, HockeyApp, Google Mobile App Analytics, and Amazon Crash Reports.

They all work on the same basic premise. You install usually a small framework from the vendor into your app. It sits quietly, waiting for mayhem. When things go wrong, the framework collects as much information as it can about the operating system version, device type, error message(s) and possibly a list of steps that led to the crash. That information is delivered to a server whenever a connection is available again. You will optionally get an email notification regarding the crash and will undoubtedly be able to see reports in a browser or on a mobile device that may help you identify trends and common causes of crashes.

Performance tools

Apple and Google both provide tools for developers to research and monitor app performance during the development process. Apple has a nice visual tool called Instruments built into their Xcode development suite for iOS. Developers can run their app while Instruments monitors memory usage and leaks, slow passages of code, CPU utilization, and much more. Developers and testers record a sequence of steps in their app, make

code changes to improve performance, and then play back those steps automatically in Instruments, comparing performance across runs to evaluate the effectiveness of the code change.

Energy and battery usage

Battery use is of critical importance for mobile devices, especially phones. While devices are powerful, their small size means they have physical limitations when it comes to space for batteries. It is important for your app to be a good power citizen on a user's device or risk being uninstalled. Users want to use your app, but not likely at the expense of their ability to make phone calls throughout the day. In iOS 8, Apple introduced tools in the Settings app that allow end users to see a list of apps on their device using significant energy. You do not want your app on that list!

Using the performance tools described above is a great way to diagnose features in your app that might be power hungry. There are also published best practices for iOS and Android to help conserve battery power through your development practices. For instance, limiting the use of GPS and other location services will help. Checking in with servers less frequently will give the device radios a chance to rest and conserve power.

Wearable Technology

When we wrote the first edition of this book, there was not yet much chatter about wearable technology. Now it is getting lots of attention. Since our phones are never far out of reach, in a pocket, on a belt clip, or attached to an armband during workouts, some may argue that our smartphones are wearable technology. But the term more often refers to smaller devices with sensors and varying degrees of interactivity that frequently support health and fitness.

Examples of wearable technology include: Fitbit, Nike+ FuelBand, Google Glass, Apple Watch, Pebble, and over a dozen watches from Garmin. Other examples that you may be less familiar with include CuteCircuit's HugShirt, Sensoree's GER Mood Sweater, and Ring ZERO by Logbar. There is also an emerging market for "smart" clothing and other wearable sensors to do things like support firefighters by monitoring their heart rate, temperature, and more during an event.

Wearable devices are far less capable than our smart phones and certainly our computers. They have small

processors, small batteries, and limited connectivity options. In some cases, you can build apps that run directly on a wearable device, but more often development and integration options fall into one of two other camps - data/file exchange, or a surrogate app running on a more powerful device. Fitbit falls into the first category. At the time of this writing, developers don't develop apps that run on the Fitbit. Fitbit collects data and reports it to a server. Developers can write apps that read data from and write data to that server. Third-party apps don't talk directly to the Fitbit device. The first Apple Watch fits into the second category of integration. Developers write code that communicates with and runs on the Watch, but all heavy lifting is handled by an app that runs on a user's iPhone, which must be nearby. Apple's watchOS 2.0 will allow developers to create apps whereby more work is done on the Watch, with less reliance on an iPhone in close proximity.

The phenomenon of capturing and reporting a whole host of data about individuals, again frequently for health and fitness applications, is referred to as "quantified self". Device and sensor manufacturers and software developers are constantly finding new ways to provide quantifiable data to individuals, their nutritionists, physical trainers, and physicians.

"The Internet of Things" and Apps

For decades, we have accessed the internet through our devices, including desktop computers, laptop computers, smart phones, tablets and more. Those devices are tools that connect to the internet and function under our command and control. The Internet of Things refers to the idea that ordinary objects— those that we wouldn't think of as traditional computing or networking devices— can connect to the internet and provide services, consume services, or both autonomously. They don't necessarily require a human to direct them, at least not on an ongoing basis.

Google's Nest Learning Thermostat learns your habits, adjusts its settings automatically, and can be controlled from anywhere with an app. Siemens' camera fridge allows you to capture a photo of your refrigerator contents from anywhere (e.g. the grocery store) to check inventory when you may not be able to open the door. Mr. Coffee makes a line of smart, connected coffee makers that can be controlled from anywhere to start brewing, adjust brew times, and more.

If a device is “smart” or “connected” does that mean that you can build apps for it or build apps that communicate with it? Maybe. The device manufacturer needs to make that capability available to you, generally using either APIs or SDKs referenced earlier. However, it is possible that a vendor can choose not to publish an API or provide an SDK and keep their platform closed to third parties.

Micro-location

There are many ways in which your mobile device can learn about and report its location. GPS is one of the more obvious technologies for devices equipped with that capability. It has the potential to be highly accurate, but it is also a power hog and requires line-of-sight connections with multiple satellites. An alternate method is to use cell towers to triangulate a device position, using much less power than GPS. A third choice employs wifi signals to determine location. There is nothing in the wifi signals themselves that broadcast location. Rather, a database correlating wifi base stations with geo coordinates is used to approximate your location based on the strength of the signals from nearby wifi base stations.

Bluetooth technology is a fourth option, and it provides a low-cost, low-power alternative to the other three location technologies. Bluetooth devices providing location services are often referred to as “beacons” because they emit a signal. Apple branded their implementation of Bluetooth beacon technology as iBeacons. Your phone, tablet, or other device can be configured to behave as a beacon. But it is more likely that it will be used to scan for nearby beacons.

Like wifi, beacons provide no location information inherently. A device that detects a beacon makes assumptions about its proximity to the beacon based on the strength of the signal it detects. Interference will produce flawed results. Beacons can usually emit signals about 70 meters, but long-range beacons might reach 450 meters.

Beacons are often used to let you know you are near something, even if the physical location of that something is less important. This provides some interesting capabilities beyond just identifying your spot on a global map. For example, a beacon might be placed near a piece of art in a museum. If you have an app on your device that knows about that beacon, it will detect that you are near the art and can display information about the

piece. If the museum moves the artwork to another floor or wing, the app will still work just fine because it is simply detecting that you are near the beacon, wherever the beacon happens to be. Likewise, a retailer might want an app to take action when a customer gets close to a register or a particular product display.

Bluetooth beacons are a great way to provide fairly precise indoor positioning, but they require that someone associate location information with each beacon. For example, Apple is working with malls, arenas, and other large venues to help them map their interiors so apps can provide services to visitors. Directions to food services, restrooms, and your seat are all possible in apps once beacons can be tied to a location, even indoors.

CHAPTER 4

RESOURCE PLANNING

This chapter provides guidelines for considering the financial cost of an app, and describes the responsibilities of the people involved in the project.

RESOURCE PLANNING

In This Chapter

Finances

Development cost

It is exciting to think about the features in your app, the anticipated market success or adoption by your internal team, and benefit to your organization. But thoughts quickly turn to the finances for app development.

It is a common misconception that apps must be cheap and easy to build if companies are able to give them away or sell them for $0.99. Like the mainframe systems, desktop applications, and web applications that preceded them, apps are software. Tools are constantly improving to make software development more predictable and easier in some respects, but those improvements tend to support the expansion of what is possible rather than dramatically reducing the cost of development. In The True Cost of Building a Mobile App,

the author states, “Serious mobile apps require a strong conceptual foundation, good planning, an excellent ecosystem and top-notch talent in both the design and engineering phases. Scrimp on any of these elements and you risk the value and ROI of your finished product."

The average cost to develop a mobile app is a difficult number to pin down. The range is quite wide - from free to over $1 million. According to OS X Daily, a site providing tips for Mac users, a “small” app can cost between $3,000 and $8,000, while costs for a “more complex or recognized brand app” can range from $50,000 to $150,000. So how do you know where your app will fall in that range? Your development partner will work with you to determine the cost based on a thorough review of your requirements.

There are many similarities between the process of building a house and the process of building an app. Architects and builders need to know square footage, the number of stories, bedrooms, baths, and garage stalls, etc. They also need to know the overall quality of the materials and craftsmanship you expect. The same is true for apps. At the heart of an accurate estimate lies specific project expectations, requirements, and scope. This topic is further explored in the Discover section of this book in Chapter 5.

“Value” in app development

You want to develop your app as economically as possible while achieving an acceptable level of quality, functionality, and reliability. How do you begin to discern the balance point between cost and value?

Imagine you have an app you want to develop for internal use in your organization. You sent a description of your idea to four potential development partners. Here are the responses you received:

- XYZ App Development quoted $35,000 to develop the app. In their quote, they included in-person discovery meetings, project management, design services, and incremental releases for testing. They also state clearly that any changes in the requirements will affect the cost and the timeline for completion.
- An offshore development firm quoted $17,500, design services not included. You will need to complete their template specifications document before the quote is confirmed and work can begin.
- Your inside IT department said they don’t actually do

quotes, but they were sure they could create the app, although it would be their first. The CIO was unclear about when it would fit in their schedule.
- Your very creative niece said she would build it for $3,000. She just completed an app development course in college and got an A.

Of course, this illustration is a bit of a caricature, but it points out some aspects of “value” in app development which you will want to consider. Each of the developers listed above might be the right one in a particular instance. For example, if you have an idea for a simple game you think would be fun to develop and distribute, your niece may be the best choice— and you might have a lot of fun working together. If your idea is for a non-public-facing inventory tracking tool for your sales team, which can be developed incrementally as resources are available, your IT department might represent the best value. If your app has rock solid specs and doesn’t need to be integrated with your business in other ways, the offshore option might be acceptable, but there are fundamental challenges to this option which need to be considered. If you choose to outsource, be sure to analyze the options.

Apps that are complex, that require more organic development (i.e. you expect to discover new features during the course of development), or that will have critical impact on your business deserve highly professional development in the context of a close working relationship with your development team. The fictitious XYZ App Development company, and many real development specialists like them, specialize in these type of development services you need.

David Adams, a respected developer, said “A $225/hr developer is cheaper than a $25/hour developer. I know. I’ve been both.” There is wisdom in those words. You should expect to, and to some extent want to, pay more for app development if:

- Your app is quite complex
- You value a highly engaged working relationship with your development team
  You want to ensure that best practices are followed
- You desire a spectrum of services from your development team including: design work, marketing input, and post-launch support
- Your timeline is critical
- Your app is the first in an expanding suite of apps
- The app will reflect on your organization’s public

image
- Your app involves intellectual property or trade secrets which must be securely protected

Choosing the right development resource for your app may be the most important decision you make to ensure its success. Your challenge will be to balance cost and value in a manner that suits you.

Billing for development time
Development projects are usually completed on either a fixed-price or time-and-materials basis. Additionally, revenue-share options are explored in some partnerships.

Fixed Price
In a fixed-price arrangement, you will receive a bid from your development partner to complete your project. Typically, you agree on the scope and price before work begins. It is an advantage to know up-front what you will pay for the finished product.

There are some downsides, however. Fixed-price projects require more time defining detailed requirements in the discovery phase. To deliver an accurate, reliable and firm bid, the project scope will need to be nearly ironclad before any work commences.

Fixed-price projects also come with additional project management overhead. Each time you want to modify the product in a way that will impact the scope and development cost, the developer will prepare a change order describing the additional work and the price to complete the work. This practice could result in more time spent on administrative tasks which would otherwise be devoted to development and project completion.

Time-and-Materials
In a time-and-materials (T&M) arrangement, you pay for each hour your development team works— no more, no less. You also typically pay for travel expenses and project-specific expenses the development company incurs (e.g. specialized hardware or software). Rates may be based on the type of work being done or the seniority of the person working on the project. Some companies charge a single blended rate for all project work.

The risk to you in a T&M relationship is that project expenses may exceed expectations. Your development partner should provide you with an estimate including a degree of confidence or a range so you can plan

accordingly. The advantage to you in a T&M relationship is reduced project management overhead and the flexibility to make course corrections over the life of the project.

Revenue Share

Although it is less common, some developers are interested in working for a reduced price, whether fixed or T&M, in exchange for a share of the project's ongoing revenue. This is less common because developers are usually highly skilled and knowledgeable about app development but have less expertise in your industry. You may think your app is a guaranteed hit. But you've arrived at that conclusion as the result of deep industry or customer expertise in the space your app will serve. Without similar understanding, trading immediate revenue for potential long-term gain may not be attractive to your development partner.

Some customers see revenue sharing as a way to align your interests with those of your partner to ensure project success. It's a valid strategy. If this is your goal, consider offering to share a small portion of your revenue or profits over and above the normal development fees.

Hosting fees

As you think of your app requirements, hosting may not come to mind. After all, isn't that something you do with your website and not apps? That is true for many apps. Those that are downloaded and installed from an app store and work offline with no need to exchange data with a server do not need to be hosted.

However, mobile web apps require hosting because they function similarly to a website targeting desktop browsers. Native apps will have hosting requirements too, depending on their need to exchange data with a server. Many functions of mobile apps will require a hosted server. For instance, you may need to synchronize data between mobile devices. That data will pass through a server to coordinate its distribution. Mobile professionals may be collecting data in the field to be stored in a central database for later analysis and presentation on the web. That database will be hosted.

There are many hosting options. For high-availability systems and those that may need to scale rapidly to meet growing demand, you may consider virtualized hosting services from Amazon or Rackspace. Hosting fees are based on the memory, storage, and processing

capabilities of your virtual server(s) and the amount of data exchanged with them from the outside world. Prices range from about $15/month to hundreds per virtual server. If you require features like database replication for redundancy, load balancing, and backup, you many need more than one virtual server.

Project Roles and Responsibilities

The success of development projects with a lot of players hinges on building a strong peer-to-peer relationship between all parties. A clear communication plan should be determined early on to ensure milestones are agreed upon and accountability is shared. Everyone should feel like part of one team, working toward the same goal. App development requires experience and input from a variety of key players. Here are some people that will likely be involved:

Project Manager

The project manager is responsible for leading the effort from inception to completion. S/he leads the team, tracks progress to stay on-time and on-budget, and functions as the communication hub. Assembling the right team, giving appropriate direction, and managing the working environment while maintaining a strategic focus and staying on top of deadlines and deliverables are all part of a project manager's role.

Designer

Designers are responsible for not only the visual look and feel of the product but also the entire user experience (UX). Good designers interpret written requirements and use cases in a visual way to elevate project understanding. While creating their designs, they work with developers to make sure what they produce can be implemented within the project scope.

Developer

Developers are responsible for creating the software that satisfies the project requirements. Most software development processes champion cross-functional team collaboration early and often in the project lifecycle. It is important for developers to not only understand the project goal but also have a hand in shaping it. This allows developers to identify potential pitfalls and opportunities early in the process.

Quality Assurance Expert

Quality assurance experts are tasked with ensuring there are no surprises when your app goes live. They will

evaluate every aspect of your app with a multitude of devices to identify any issues with its functionality and work with the developer to resolve them.

Project stakeholders

A common definition of a stakeholder is any person who is actively involved in the project or whose interests may be affected by the execution or completion of the project. Stakeholders are generally people within your organization, but in some situations (e.g. start-ups), they could be people who have a vested interest in your company but don't work for it.

Other possible stakeholders include executive team members or leaders from various functions throughout your organization (e.g. IT, product development, and marketing). To ensure all stakeholders have been included in the process, it is important to evaluate whose support, or lack thereof, might significantly influence the success of the project. Turning stakeholders into project champions early on can reduce project churn and minimize friction.

While it's important to know who all stakeholders are, you may not want all of them on the core team. The smaller the team, the more nimble it will be. Some stakeholders simply need to stay informed about the project's progress, while others provide critical input to project success.

End users

As you think about key stakeholders, the most obvious one is the end user of your app. Hopefully you have been thinking about this user from the start and using insights from your market research to shape the requirements you outlined in the design phase. We call them out again here, though, to remind you that they must remain in the forefront of your mind throughout the project.

Whether your end users are colleagues in your organization or customers, their voice is one of the most important in the development process. Listen to them directly whenever possible, include them on the core team if you can or, at a minimum, use them to test your app and provide feedback as it comes to life.

Intellectual Property

Code ownership

Assuming new code will be written for your project, you will need to determine the appropriate way to handle the

ownership and/or licensing of that intellectual property. You should always consult with an attorney to determine the best approach for your circumstances. Here are some options you may consider.

You Own It All

Most customers start with the expectation that they will be the sole owners of all code that becomes part of their app. This is a reasonable expectation and one possible outcome. But requiring all original code to be entirely client-owned may lead to a more expensive project. If you contract with an experienced development partner, they will likely have access to a library of previously developed, thoroughly tested code which will help them develop more robust, complete solutions for you more quickly. To accomplish your goal of owning the entire completed work product, the developer will have to forgo use of their framework and start with a blank canvas.

The Developer Owns It All

This is rare. But if you are working with a developer who has already built a solution that is nearly identical to the one you seek, you may find yourself paying much less to have it configured for your needs rather than funding new development from scratch. If this is the case, the developer is likely to retain ownership of the code, granting a use license to you.

Joint Ownership/Licensing

A best-of-both-worlds solution involves a mix of the first two options. The developer is able to retain ownership of their growing library of utility code that travels from project to project and serves to save you time and increase the quality of your product. This is sometimes referred to as “background technology”. The developer grants a license for you to use the background technology in your project but usually restricts you from selling, open sourcing, or publishing their code.

In turn, you own any new code that is written specifically for your project. This usually includes your business logic, or the “secret sauce” that differentiates your product in the market.

CHAPTER 5

THE PROCESS

This chapter describes the steps for creating an app from start to finish and what to expect from the development process.

THE PROCESS

In This Chapter

Process Defined

It's important that you and your development partner have a shared understanding of the development process before you get started. The details of that process obviously differ somewhat by company, but in all cases, the goal of the process is to ensure that the partner builds what you expect, for the agreed upon budget. Process aids in managing scope, providing project status, and exposing dependencies.

The best process allows for change and understands that flexibility is healthy. The word "process" can be perceived as inherently rigid. Instead, think of it as a general guide for how you intend to proceed. You will first give thought to what you intend to build (discovery), then capture those thoughts so they can be shared and refined (design). Next, during the build process, you will see the product emerge (development and testing). Then, after the product is deployed, you should expect to have post-launch plans for enhancements, maintenance and support.

Discovery

During the discovery phase, also referred to as analysis,

the aim is to understand your product vision and objectives. The kick-off is usually an in-person meeting to discuss the idea, ask lots of questions, listen carefully, and record accurate responses.

The end goal of this meeting is to be able to visualize the entire scope of the app and its diverse set of users and uses. There are various ways to accomplish this, but our company uses a methodology called User Story Mapping to layout the needs of the users and ensure that the various use-case scenarios line up with the product objectives.

We can't emphasize enough that an accurate understanding of the user must inform your design! Such understanding comes from the market and customer development work we outlined in Chapter 2. By clearly outlining and validating your customer assumptions, you are able to focus on building what they will actually buy. This will spare you from building features a customer would never use, or possibly introduce new features you never anticipated.

After the user stories are mapped out, there is a second level of planning questions that need to be addressed. These questions tie to the technical considerations we outlined in Chapter 3, and include items like:

- Which platforms (iOS, Android, web, etc.) are right for your app?
- What device form factor makes the most sense?
- Should the app integrate with internal or third-party sources?
- Will the app function offline?
- What analytics are useful to capture?

When these questions are answered, feature sets can be created and grouped into releases that define the timeline, build the user story backlog, and enable you to plan effectively.

Along with members of the customer team and a Project Manager, at least one developer and one designer participates in the discovery phase. Including a member from each discipline (design and development) adds a broad perspective and leads to an easy transition to the other phases of the process.

Although the goal of the discovery phase is to define the overall scope of the project, it does not mandate that the product be designed and built in one large effort. Having

a clear vision of the destination will help to chart your course. But we still recommend tackling the effort in small chunks, releasing work-in-progress frequently along the way— ideally to real users. That will allow you to collect user feedback early and make adjustments before building a large system based on flawed assumptions that is difficult to change.

Thoughts on gamification

As we talk with clients about their vision for an app, there are three terms that have emerged in many of our conversations over the years: fun, sexy, and gamification. It's worth taking a moment now to understand these terms better and understand how such concepts fit into your design requirements.

You want your app to be fun to use. That is a terrific goal. Apps that are fun to use are likely to be used more often. You want your app to be sexy— at least some of our clients do. Apps that invest in high-end design are striking and do create a positive first impression. You want your app to draw people back, and that's important. Gamification refers to the notion that adding elements of game-play will make an app more engaging, and that the options to compete with a community or earn achievements will encourage regular use.

There is a lot of science to support employing techniques that you see in games to encourage adoption and engagement in non-gaming apps. However, that doesn't mean that you should expect similar adoption and engagement in your app simply because you use these techniques. Don't be surprised if you don't see much behavior change when you award achievement badges in your employee timesheet app. Information technology research company Gartner has stated that they believe that 80% of gamification applications will fail if not designed correctly.

The key part of utilizing any of these concepts is that they must be part of your design from the start. Too often, they are seen as an add-on— something that can be applied as a final step in the process— to increase the appeal of an app. "That how-to plumbing app you designed and developed for us meets all of our stated objectives. Now, can you jazz it up a bit?" Probably not.

This aspect of your app design and development needs to be thought about as you might think about fitness. There are no shortcuts. Good nutrition and exercise are

hard work, but produce results. If you want to build a fun app, invest the time and effort to really know how your target customers will use your app to improve their lives. Then delight them by delivering something that does just that. They will be engaged because you've thoughtfully addressed their needs, not because you slapped that adorable animated icon into the app to make it more fun.

Design

Users have high expectations regarding the form and function of their apps. Design plays a role in both of those. The responsibilities of design in the app process are often split between user experience (UX) and user interface (UI). You'll find more details about each below.

User Experience (UX) Design

When a design team actively participates in the discovery phase, they can quickly and effectively move to the design phase because they understand both user needs and the business objectives of the project. Design includes the highly-important element of User Experience (UX). Understanding how people will use the app, and the workflows which make navigating and using the app simple and intuitive are intended to combine to create an overall feeling of user satisfaction.

The first visual deliverable in the project is annotated wireframes which behave as an interactive, semi-clickable prototype. Wireframes consolidate ideas from the discovery phase into something tangible and easy to visualize. Wireframes are simply a representation of the skeletal structure of an application, often compared to blueprints for a building. The purpose of wireframes and the focus of this phase is to understand what the screen does and not what it looks like.

Strong collaboration at this stage creates a solid foundation for moving your app forward. Wireframes can also be shared with potential customers and internal stakeholders to elicit feedback and buy-in. With wireframes, you begin to see your product coming to life. Invariably this leads to a more focused target for development.

User Interface (UI) and Visual Design

Wireframes evolve. The initial set of wireframes will be revised, refined, and recast based on your input. Once wireframes have been approved, the design team will begin designing the app. The team will gather branding guidelines and relevant assets from the client. This is the

stage where interactive components such as icons, links, and buttons are chosen to define the way the app will respond to user inputs and actions, and visual elements such as color, graphics, and typographic styles are woven into the wireframes.

The user interface (UI) of a software application includes every point at which the user can interact with the software, whether that is clicking a button, checking a box, or scrolling down a page. Each and every possible action and response is carefully considered by the Designer in order to make interacting with the app easy, intuitive, and effective.

Visual design focuses on the aesthetic appeal of the app, and impacts users' emotional attachment to the tool you are creating. The app's visual identity should also be informed by your brand's personality, so it's important to involve your marketing team when evaluating options. See General Considerations in Chapter 7 for more details. It is through the careful consideration of effective UI and visual aesthetics that the finished designs for your app's interface will provide an exceptional user experience.

Once you have the product of the discovery and design phases, a more accurate budget and timeline forecast can be created.

Development

The development phase is when the functions and features molded during the design stage become reality. The software engineers have participated in the previous project phases, so they are up-to-speed and ready to create the app. The engineers begin to code using iterative programming cycles to develop and test functional prototypes.

While development specifics differ by company, we believe using an iterative model that allows for adjustments to the app during development is the best because it:

- Provides room for improvement in subsequent iterations based on lessons learned from previous iterations.
- Breaks the overall project into manageable feature sets, with each set building on the previous.
- Allows flexibility in accommodating new requirements or changes to existing features.

Testing

Testing is sometimes considered the last little step before the app launches. But, like many of the steps along this not-so-linear path, testing is something that needs to be integrated from the beginning, done continuously throughout the process, and fully executed before the app enters the real world.

After the wireframes are approved, your partner's quality assurance (QA) team will likely begin writing test cases. Test case creation allows the QA team to be up-to-speed on the scope of the app and ready to begin testing when each feature is ready. This continuous, iterative testing helps avoid large surprises at the end, since each component gets evaluated along the way.

The QA team also validates that applications behave correctly across whatever devices, browsers, and platforms you are targeting. They ensure that the app can scale to meet the demands of the real-world user base.

It is vital to have dedicated testing staff on your internal team as well. Testers should be people who have been involved from the beginning and understand the desired functionality. Depending on what you are creating, you may want outside testers as well— people who can give you a completely unbiased viewpoint. While you may not be testing components as regularly as your partner, you should be able to preview your app for testing early and often in the process. It would be a mistake to wait to evaluate it only at the end.

Deployment

The entire development process leads up to the exciting moment of deploying the app! Deploying the app from a technical perspective means that your app is fully functioning and ready to be installed onto user devices. Getting the app into the users' hands may mean you are releasing the app to one of the app stores— either widely to the general public or to a smaller population of beta testers. You may also make it available to users within your company via your intranet or via some other corporate app distribution channel. There are many options, so we've dedicated Chapter 6 to discussing app distribution details.

Deployment from a technical perspective should align with "launching" from a marketing perspective. Regardless of whether you have built an internally-facing app, or an externally-oriented one, potential users will

need to be informed that something new exists and that it's worth their attention. The planning and execution of your marketing plan should be concurrent with technical development. An introduction into some marketing concepts is provided in Chapter 7.

Monitoring

While launch day is the key milestone for which you and your partner have been aiming, the work rarely ends there. You will want your partner available for the days and weeks immediately after launch, in case of surprises in performance or even questions about functionality. They can help you to monitor engagement with the app or evaluate other performance metrics to ensure the app is off to a strong start.

In addition to these short term needs, you will want their partnership in the evolution of the app. If you've built an app that's truly valuable to your end user, the app will need to change and grow with them over time. In Chapter 8 we provide an overview of some ongoing considerations for the app you have brought to life.

CHAPTER 6

DISTRIBUTION

The app is built and you are ready to release it. In this chapter, you will learn about your distribution options for market-ready apps.

DISTRIBUTION

In This Chapter

If the goal is for your app to be distributed publicly via Apple's App Store or Google Play, you may need someone to help you navigate those waters. You will need developer accounts with those outlets (even if you aren't doing the development work yourself) and will need to coordinate the launch with your marketing team. Planning should start well before your intended distribution date.

For internal enterprise distribution you don't need to worry about potential issues with a store or marketplace, but you may need help with distributing the app to your users in the field. Additionally, distribution and the communication plan surrounding it will need to be coordinated.

Distribution Options

The storefront you use to get your app into the market depends on decisions made earlier in the process about the form and function of your app. For instance, if you decided to focus on Apple products, then you'll use Apple's App Store. Alternately, Google Play and Amazon represent options for Android apps. There are four notable distribution options for your finished product: app

stores, ad-hoc distribution, enterprise distribution, or via the web.

App stores
People are most familiar with this option. You submit your app, it goes through a review and approval process, and is then made available for consumers to purchase. The proprietor of the store generally charges a percentage of your app's sale price to appear in their store. Typically, there is a 70/30 split with 70% of the sale price going to you and 30% going to the store. Payments are made via direct deposit on a monthly basis.

iOS
Only one app store exists for iOS: the Apple App Store. You need to obtain an Apple Developer account in order to distribute the app under your name. This process can take a bit of time, so plan to complete it early.

Apps typically take 1 to 2 weeks to be approved in Apple's App Store. The time varies based on the complexity of the app and the backlog of apps in the Apple queue. Apple reviews to ensure that the app works as advertised, honors copyrights, doesn't use any restricted features, and more. That review process can seem daunting the first time, but Apple publishes their review guidelines so you know what to expect. We have found that the turn-around time on reviews of subsequent submissions of versions of the same app tend to be shorter than the first release.

Android App Stores
There are two main Android app stores: Google Play and Amazon Appstore for Android. You will need to register for developer accounts for both of these stores, and again, should be completed long before your target release date. Google Play is the default for all devices except the Amazon Fire line of phones and tablets. Android apps historically do not go through a human-powered review and approval process. However, in Spring 2015, Google announced that it was introducing humans to their review process. Despite that change, Android apps on Google Play are usually reviewed and ready for sale within hours of submission— not days.

Ad-Hoc
Ad-hoc distribution allows developers to build and distribute apps directly to users. This type of distribution is intended to be used primarily for beta testing and not wide-scale distribution to customers.

On most platforms, this type of distribution is restricted because app store operators prefer customers purchase apps from their marketplace. For example, to distribute iOS apps ad-hoc, developers are required to specify the devices on which the app is allowed to run. The app can then be shared with up to 100 other iPad, iPhone, or iPod touch users.

Enterprise

Corporate entities building apps for in-house use may opt for enterprise distribution. It is similar to ad-hoc distribution in that you are able to distribute apps directly to users, but you do not need to specify the devices on which the app will run. With an enterprise license from Apple, app owners are responsible for limiting distribution to employees and users within the organization only. There is no revenue share or per-user charge for enterprise distribution.

Web

If you build a web app, you may avoid the traditional distribution and installation process described above. Instead, users will run your app by visiting a particular web URL. On most touch-based mobile operating systems, users can opt to install a web app by adding its icon to the home screen on their device. The icon is indistinguishable from native app icons on the device. When the user taps it, the browser on the device takes over and runs the web app.

MDM: Mobile Device Management

Large enterprises have sophisticated Mobile Device Management (MDM) needs. Some companies purchase and issue all equipment to their team while others support “Bring Your Own Device” (BYOD) whereby users can purchase the device of their choice and use it personally and for work. In both circumstances, companies will want to control access to the corporate network from mobile devices, control distribution and installation of apps on devices, be able to remotely wipe a device if it is lost or stolen, and revoke access to resources when an employee moves on.

MobileIron and AirWatch are two popular commercial MDM packages. If you are building an app for distribution within your own large company or selling your app to large enterprises, it is likely to be distributed to end users via an MDM. If that is your primary market, it will behoove you to learn more about it. If not, just know that you will

need to coordinate with the IT organization of the enterprise to get your app to your colleagues.

CHAPTER 7

MARKETING YOUR APP

The adage, "If you build it, they will come" does not hold true in a marketplace flooded with apps, books, and games. This chapter describes some basic principles to get you started in marketing your app.

MARKETING YOUR APP

In This Chapter

In 2014, over one million apps were available in the App Store, but very few garnered significant revenue. With the proliferation of apps, even the best products need concentrated marketing attention to rise above the noise. Your marketing team has ideally been involved since the beginning (see Chapter 2), helping to identify needs and define objectives. As your app gets closer to launch, their work intensifies to get the word out about the amazing app your company has put so much energy and brainpower into producing.

We'll begin this section reviewing some strategies and plans that need to be put in place prior to making consumers aware of your product. We'll then discuss how to create awareness with potential customers, and wrap up with some thoughts on how this process differs for an internally-facing app.

General Considerations

Prior to beginning your app development project you took the time to clarify the value your app could bring and determine how creating it might fit into your broader business strategy. Now is the time to dig into the details and figure out the best way to convey the benefits to your

target audience.

Define product branding
Branding is a complex subject, but it can be likened to personality— it's the unique essence of who you are. It sets you apart from the competition and helps users fall in love with your product. Your product branding efforts will vary depending on whether you are integrating with an existing company brand or starting from scratch.

Brand creation is beyond the scope of this book, but we mention it here simply to encourage you to pause and make sure this important framework is in place. Many people want to rush right into the tactics of marketing a specific thing, but your brand involves more than just the nuts and bolts of what you are selling. A brand encompasses non-tangible consumer concerns, giving insight into the values of the organization. While marketing may draw in a sale, it is often the promises of the brand that create loyal customers.

Create a name and visual identity
Once established, your brand identity ought to come through in several elements particular to the app itself: the app name, logo, icon, and splash screen. The visual look and feel you define here should be connected to your company at large, and also carried forth into other marketing collateral particular to the app.

App Name
Naming is not an easy endeavor, but it's important to spend adequate time and resources to select a compelling name. The name should stay with the product indefinitely so it's ideal for it to be easy to say, remember, and spell. It is difficult and expensive to change your name later without losing equity. According to the naming agency Igor, a name can be descriptive, invented, experiential, or evocative. Learn more from them by downloading this guide.

It's important to note that there are officially two names associated with your app— the Bundle Display Name and the App Name. The Bundle Display Name is about 11 characters long and is the name that appears under your app icon on a home screen. This name is how people will likely refer to your app, so avoid choosing too generic a name or you could lose customers due to confusion. The App Name is longer (max 255 characters) and appears in the App Store at the top of your page. In this name you can include several keywords that describe the

functionality of your app. It’s great to take advantage of this opportunity, for it improves your search results, but caution needs to be taken to avoid sounding spammy.

Logo
A logo is an essential element of your app's brand. Logos can be simple or complex, and include a logotype (text only), a logo mark (icon), or both. It is common to have several versions of your logo for different uses. Your logo will often be viewed on a mobile device, so be sure it looks clear and high quality at a small size.

Icon
Not to be confused with a logo, the icon is the graphic that will identify your app when you go to market and on user's devices. Icons can be anything from a real-world object, to a logo mark, to an element from the app's visual design. The app name, in text, is almost always displayed near the icon so it is not necessary to include your app's name in the icon design. Keeping text out of the image also makes it easier to localize your app for different languages.

Splash Screen
Visual design is important all throughout the app, but there is a unique opportunity for branding with the splash screen. The splash screen is the full-screen image that appears while your app is loading after being opened. Since it is only visible for a short amount of time, minimal text is best. The look and feel of this screen should support your brand personality.

If there is a chance your app could grow into a suite of apps, consider creating an identity which is flexible enough to show connection among each app in the group. Names that sound similar, icons that share a uniting feature or effect, or a coordinating color scheme could all help in creating an “app constellation.”

Determine a pricing strategy
Pricing is the most important choice you will make relative to the long term success of your product. Your ability to ask a higher price increases when you’ve done good market research and, as a result, have built an app experience that is unique and valuable to your intended audience.

The price of the app, though, does not need to be the sole revenue creator for your product. In many situations, the app’s price is part of a wider mix of revenue

generating options such as in-app purchases or advertisements, subscriptions, and service fees. In some cases, the revenue gained from sales of the app itself may be secondary to the revenue generated by in-app or after-purchase sales. It is important to understand these options and how they will impact your marketing strategy.

The more you understand about market demand and how much your audience will be willing to pay, the better off you will be. You want a price that will not deter people from checking it out, but one that is high enough to reflect the product's value and maximize revenue. Anticipated sales volume is another key consideration, as a smaller market size may require income per user to be higher to hit profitability markers. Keep in mind it is much easier to drop prices than to raise them post launch. If you feel a lower initial price is needed to encourage trial and build your customer base, be creative with things like introductory promotions, free use for some time period, refer a friend bonuses, etc. These options allow your stated product price to remain higher while you determine the sweet spot.

The news blog, Mashable, provides a helpful overview of pricing strategies in their article "How To: Determine The Right Price For Your Mobile App."

Create internal awareness

Even if the app you created is designed to be bought and used by those outside your organization, it will still be necessary to educate key groups within your organization. This can range from simply ensuring your customer service representatives can answer questions to a full-blown internal launch that educates your sales team on how to sell the app. Before you start telling the world about your new product, make sure your internal team is ready for whatever role they need to play.

Invite consumer feedback

In Chapter 8 we discuss usage analytics in more depth, but it's worth noting here that obtaining and analyzing customer feedback in an on-going way is an important facet of marketing your app. Direct feedback enables your customers to help drive your product roadmap, and gives you more confidence that new releases (or entirely new products) have value. It's important to determine where you will be "ready to listen" before your app is launched.

There are various places where you can engage your

customers and request input, but inviting comment within the app is one we strongly encourage you to consider. Not only is it easy for the user, but it also provides an opportunity for you to hear and hopefully address frustrations before they hit social media. You may also gain additional context automatically so you know what a user may have done (i.e. struggled with) just prior to providing the feedback. Custom programs can be built, or you can use in-app feedback tools provided by software vendors. Be sure to get your development partner's advice on where and how to solicit feedback, because doing it poorly can actually frustrate your customer even further. This article offers some good perspective to consider.

Attracting Attention

Once direction is established for the considerations just outlined, you can begin to create your marketing communication tools. The process doesn't need to be completely serial in nature, but the aforementioned strategies should be sketched out before you start creating collateral. The digital marketing age is filled with a myriad of promotional options, and in some ways, there are too many choices. We will outline a number of marketing platforms below, but that does not mean you should employ them all! Consider whether or not the option will be an effective tool in reaching your customer segment and meeting your specific objectives.

Create an online presence

Although all marketing platforms will not apply to all apps, some level of online presence is definitely necessary. Two important pieces of online real estate are your own website and your page within the app stores.

Your own website will typically be the central hub to which you send people to learn your story. Sites vary widely in content, but at a minimum, be sure you present a clear pitch explaining why your app is awesome! Promo videos are great ways to provide a tour of your app in a fun and engaging way— and they can be repurposed in all kinds of other places from social media to blogger invitations. It's also critical to have a clear call to action. If your site is up pre-launch, include a subscription form to capture interest. After launch, include badges from the mobile app stores to make it easy to download your app. Other relevant content might include user reviews, awards won, social media links, company information, press kits, etc.

Maintaining a blog on your own site is also worth considering because it not only can be leveraged as a way to promote your app, but it's another way to draw people back to your site on a regular basis too. You can begin to develop followers of interest (even before your app is ready to launch) by building and sharing great content about your app's niche/industry. Curating key content from others in the space can help establish your thought leadership, as well as develop relationships with bloggers who might then share your posts in the future.

Regardless of the content you include on your site, take time to learn about search engine optimization (SEO). Many people learn about products and companies by doing a search for information using one of the major search engines. Understanding how these engines work, and crafting your story in a way that enables the engines to know what you offer, makes it possible for interested consumers to find you. There are numerous factors that play into SEO, from how your site is written and structured to how often others refer to your expertise. Moz.com offers a great overview in The Beginners Guide to SEO.

The app stores offer another piece of online real estate which you can largely build as you would like. Because many people search the app stores in the same way they search the web (e.g. using keywords in the store search box), it is worthwhile to make your presence in the stores as strong as possible. Sometimes called "App Store Optimization," the principles are basic and align with what we've been discussing thus far:

- Choose your app name, icon, and keywords carefully.
- Write a compelling app description.
- Include your most visually engaging screen shots and write captions to go with them.
- Pursue positive app ratings from your users.

Per our earlier comments on branding, make sure the visual look and feel of graphics, as well as the tone of voice of your copy, aligns across your site and the app stores. Regardless of where your consumer meets you, they should be able to recognize you by your brand identity.

Drive traffic

An online presence is only useful if people come visit! There are some things you can do on your own, like search engine advertising (SEA), hosting a webinar to draw attention, or initiating an email campaign. However,

your reach will be limited unless you extend your network and get other people talking about your app as well. Some high impact options include:

Partner Sites, Blogs, and Forums
If you are working in partnership with other companies, encourage them to link to your site in some way. This could be as simple as a badge on their homepage, or a text link to you in a relevant part of their site.

A blog post or article on a well-known independent site can also be incredibly helpful for spreading the word about your app. There are a number of blogs, such as cnet.com or lifehacker.com, specifically dedicated to reviewing new technologies. These can be valuable to pursue, but are certainly not the holy grail to your success, as many of them primarily reach early adopters. More of your audience may be influenced by blogs that pertain to the industry which you serve. For example, if it's a travel app, get written up by blogs, publications, and newspapers that serve travel interest groups. Starting small with local bloggers is also worth considering, as their content may be curated by others who are better known.

Posts in a more technically-oriented community or forum can also generate excitement, particularly if you've built something that employs a new technology. Follow and like people and pages that talk about app development, app marketing, great apps, etc. Get to know who has influence, and who shares interesting content you can learn from.

In all cases, reach out to the blogger individually, with a thoughtfully crafted email that succinctly conveys what your app does and for whom it's intended. Apptamin shares some great insights on how to write a successful pitch. If your budget permits, consider using a service like MuckRack to put you in contact with the right people.

Consumer App Reviews
For better or for worse, ratings in the app stores are also influential in getting others to choose your app above others similar to it. There are many broken aspects to the review process, but they are still a first indicator of value that many people use.

As a result, it's important to put your best foot forward and seek reviews that tell your story well. Find ways to get your app into users' hands by doing things like

offering an early-adopter discount. Use an in-app feedback SDK to gather input, then request they review the app on the relevant store if they love it and to tell you if they don't!

Social Media

You could consider social media part of building your own online presence— because it is! We've included it in this section, though, because its value in many ways comes from the role it plays in keeping momentum going on your behalf.

Social platforms are just that, social. They create a sense of community among participants and enable people to share things they are passionate about with others. Through the multiplying power of friend networks, they can be instrumental in spreading the word about your great new app. This in turn impacts traffic to your site, your ranking in search engine results, and ultimately sales in the app stores.

There are an incredible number of social platforms— Twitter, Facebook, YouTube, Pinterest, and Instagram to name a few. Each has its own strengths, but all require significant effort and content to keep followers engaged.

Before launching into any one medium, take the time to develop a social media strategy that thoughtfully leverages particular platforms to support promotion of your app. And remember that social media isn't just what occurs outside of your app. Consider building social sharing into the fabric of the app itself, encouraging users to brag to others as they accomplish certain goals.

Of course, don't forget that the visual design of your social profiles should align with the branding of your company, and your marketing messages should be consistent. Likewise, be sure the tone-of-voice in any online social interaction complements that of your overall brand. This practice builds trust and increases brand awareness.

Other Ways to Connect

While digital media offers numerous options for promotion, there are a variety of "unplugged" options as well. Sponsor a booth or a seminar at trade shows, conferences or other relevant exhibits. Host a Meetup in your area. Write a white paper on a topic of industry interest. Enter your app into consideration for any relevant award or contest. And, of course, don't forget traditional news/media/public relations outlets that speak to your

consumer. Most newspapers and magazines have both print and digital versions, and your presence in one often means you can get exposure in the other as well.

Launch day

The above ideas are relevant in an ongoing way, but it's also helpful to have them come together on launch day. This is a time to make as much noise as possible.

Let everyone that showed an interest, gave you feedback, and helped you in some way know that your app is ready. Prepare a special email to go out to those in your personal database, and post to whatever social media platforms you have determined to be relevant. Create a special blog entry on launch day to share the good news, and if possible, get commitments from other bloggers with whom you've been in contact to do the same. Seek similar coordination from newspapers and magazines. While the print version of a magazine might not exactly fit your timing, many post daily digital updates. Ask any of your contacts to blog, email, tweet, or post about you too.

While not required, it's worth considering if you can offer any sort of special promotion in association with your launch. People are often more willing to repost information if they are sharing a deal with others.

Marketing an Internally-Facing App

If you are among the many businesses considering app creation in order to improve an internal process or system, some of the items outlined thus far in this section will not apply to you. Nonetheless, for larger organizations in particular, it will still be crucial to think about how to get users to buy into the value of using a company app.

One of the key ways to ensure adoption is to gather broad business buy-in before you begin. While your leadership team undoubtedly sees at a high level where improvements need to be made, that doesn't guarantee they have a full picture. Establish a cross-functional team from the start who can appropriately represent any function that might be impacted by use of the app. Interview stakeholders as you would external customers, doing similar research to ensure you understand their top jobs, current problems, and possible opportunities.

As the app begins to take shape, identify key influencers and use them for beta testing of the product. It's important to work iteratively— don't just throw a product

"over the wall" at the end of development. Further, as these evaluators become excited about the benefits the app brings, encourage them to share their enthusiasm simply via word of mouth or through company chat boards. This will help to establish some grassroots interest and anticipation in advance of your more formal communications.

To this end, it's often helpful to share your intentions early and often, particularly if you anticipate users needing to make significant changes in how they do tasks. Introduce your plans at a meeting where employees can ask questions, then take advantage of email, company discussion boards, print communications, etc. to share updates.

As development progresses, the marketing team can be focused on creating materials and/or developing seminars to help train people on how to use the app and get the most from it. An official launch day is still relevant, although it might be several different days if you are rolling out the app in a staged fashion to different user groups. Regardless, set aside a time to show how the app works, and sell your internal users on the benefits of adoption.

And don't forget to set up systems to obtain feedback from users! This is just as important for internally-facing apps as it is for ones being sold to consumers.

CHAPTER 8

ONGOING CONSIDERATIONS

The app has launched. It's being used. What now? As you might expect, your work will continue if your product is successful. This chapter includes items to consider to keep your app running smoothly after the first version launches.

ONGOING CONSIDERATIONS

In This Chapter

Stay Relevant

We'll begin by stating the obvious. Make sure the essence of what you have created stays relevant to your users. If you create a content-driven app, you must plan for the maintenance and upkeep of the content as well as the app's functionality. The level of work required will vary by app, but we can promise you that very few apps have content that is completely evergreen. A streaming music app requires you to stay on top of new releases and continue to curate selections. An app that pushes fitness tips to its users as part of its value proposition can't suddenly stop doing so. In short, users will notice if you neglect your content and they will move on to another app if you are not continuing to offer what you initially promised. Your users purchased your app because it met a need for them. As their needs evolve, so too should your app.

Stay In Touch

No matter how intuitive it is, users of your app are likely to have questions about it at some point. Consider how you will support them. There are a variety of options to choose from. We've listed some below, arranged from low-touch to high-touch. Industry standards trend toward

the low-touch options, often because it is impractical to provide high-touch support to a broad customer base.

FAQ
You may be able to address common inquiries with a list of Frequently Asked Questions (FAQ) on your website. That is a self-service option for your users. If they can find their answers there, it will save both of you time. However, you should plan to handle customers with questions not yet found in the FAQ - perhaps with one of the other options below.

Support form
You can add a form to your website through which customers can submit freeform text questions and comments to you. There are third-party services like Zendesk that aim to help you collect, organize, and respond to such feedback, but you can also simply have the form send you an email message and use email to respond to the customer.

In-app feedback
Rather than have users navigate to the appropriate support services on your website, you can add features in your app to allow them to contact you directly from within your app. Again, you can receive that feedback via email or through a third-party service.

As we've already mentioned, there are benefits to collecting feedback directly in your app as it pertains to marketing and development. In addition, doing so means one less step for a frustrated customer to get the help they need for a problem they have encountered. Collecting feedback in the app also benefits your ability to fix the issue because you will likely be able to capture some context about what the customer was doing just before contacting you. This is especially important because users often have a very difficult time accurately describing (or remembering) what they were doing when they ran into trouble. Providing automated contextual information will allow you to get to the heart of the matter more quickly.

Phone support
It is less common to support mobile app customers over the phone, but you can certainly do so if you have the staff and resources. Consider how you will support users in different time zones, potentially around the globe, and users who speak different languages if you have translated your app to support multiple languages.

Use Analytics

One of the best ways to ensure product success is to routinely improve your product. As we've mentioned already, it's important to provide venues to hear from customers who are using your product. In addition to listening to their experience, you can also utilize usage analytics to analyze the more technical aspects of how people are using the app.

You can develop your own custom analytics platform to capture usage patterns within your app. You will decide what metrics you want to track, how to visualize that data (spreadsheet? pretty charts and graphs?) and then build out that custom solution. Or, you can find a platform that already exists that does most or all of what you're looking for. There are a variety of different existing services that provide a way to gather and view app analytics. Google Analytics (or "Universal Analytics"), Amazon Mobile Analytics, and Flurry Analytics are some of our favorites due to their ease of use, quantity of data reported, and clean UI.

Those analytics platforms are available for free until you reach very large volumes of data you are capturing. And to start tracking, you only need to insert a small bit of code into your app. That's it. From there, the world (of data) is your oyster. You're able to log in to a web-based dashboard where you can view, manipulate, and analyze your app usage data to your heart's content. Some of the basic items that the services track include new and returning users, sessions, session length, device types, carriers, user paths, user location. All of the services also allow you to create custom events, in case their standard set of data isn't targeted enough.

Once you have selected a platform and can see your analytics data coming through to the dashboard, it's up to you how you use that data. For many people, focusing on one or two data points used to chart growth and changes over time is enough insight. For others, tracking a wide variety of data, slicing it up and segmenting it for deeper analysis is the next step. Many people look at a particular statistic (e.g. returning users vs. new users), notice patterns (e.g. "Why are my returning user numbers declining?"), and make changes to their app based on their findings (e.g. "I'll provide incentives in the app for returning users.").

One key piece of advice: be sure to know what data is

most important to you before you dive into the data. Screens full of data aren't going to be meaningful to you unless you know what is valuable to examine. For example, the providers of a content-rich app might be more interested in how many repeat users come back to the app to digest content, whereas a retail app provider may not be as concerned about the number of returning users as they are about purchasing conversions.

Whatever analytics platform you choose, be ready to gain some great insight into how people use your app, and also think about how you might use that data. The actual data collection and reporting is the easy part. Deciding what to do with that data is the key component to using analytics successfully.

Watch for OS Upgrades

In addition to staying abreast of insights from your consumers, it's important to remember that the mobile operating system (OS) vendors are not standing still. Historically, Apple has released a major new version of its operating system annually. Google has done so less frequently. Microsoft and RIM release even less frequently. Your users will upgrade the operating system on their device (if they can) and even purchase new devices to take advantage of new OS features. They expect you and your app to keep up. With new OS releases, the vendors introduce new features and usually try to clean up and simplify tasks for developers. In doing so, sometimes they introduce new ways of doing things that are incompatible with the code you have. You may need to make code and interface design changes to your app, just to keep up with OS updates, even if you don't add any new features.

Apple supports their current OS version and one version previous, and they encourage developers to do the same. They don't want to be bogged down supporting old devices and users who are reluctant to upgrade. The Google market is much more fragmented. We suggest supporting two to three major iOS versions and looking for a sensible break in the Google market where you can cover about two-thirds of the installed base.

Tied to this, you will need test devices that run each of the operating system versions you claim to support. (Don't rely only on a simulator for your platforms. They aren't the same as real devices.) Your OS support strategy will influence what hardware you need to have on hand to evaluate performance.

If needed, you can find used devices on eBay, Amazon, or other marketplaces. Or you can consider leasing devices with a refresh cycle that is in line with your OS support policy. For instance, if you only support the two most current versions of iOS, leasing devices for 24-30 months should work well.

Make Enhancements

If your app is a success, there is little chance that it will remain at version 1.0. Customers will make requests or log complaints. You will find new ways to improve their app experience based on the analytics you've run. New services will be introduced by the OS vendors and you'll want to integrate with them. You'll find simpler and more intuitive ways to help your customers accomplish their tasks. All of these desires will apply pressure to maintain and enhance your app.

However you decide to enhance your app, you'll need to plan for it by setting aside budget and work hours in advance so you can nimbly respond to customer requests. Depending on the scope of the change, the enhancement requests could be handled by a simple ticket-tracking software, or it could be large enough to warrant a minified cycle of new requirements, wireframes, development, and deployment.

As a rough guideline, we suggest to our customers that they plan to invest roughly 15% of their initial development budget into ongoing maintenance. If your app is quickly adopted or has a particularly active user base, you can expect that percentage to rise considerably, but 15% of the initial development budget is a good place to start.

How often will this be necessary? Multi-year release cycles used to be the norm for desktop applications and operating systems. That timeframe won't cut it for mobile users. Quarterly, monthly, or even bi-weekly releases are not out of the question. Know your users and keep them happy!

CHAPTER 9

CONCLUSION

CONCLUSION

Our goal in writing this book was to take some of the mystery out of the app development process. There are many reasons why someone who has never thought about tackling a software development project might now do so, given the increased use of mobile devices in all spheres of life.

Such a task is daunting though, without at least a basic framework for concepts and understanding of key terminology. We sincerely hope this book provided such insights around what is involved, and you now feel equipped to take the next step. There are numerous qualified developers around the country, but if you don't already have a partner in mind, we'd be delighted to help turn your idea into reality.

About InspiringApps

InspiringApps is a mobile and web app development company in Boulder, Colorado, serving clients in the greater Denver area and around the US. Since our inception in 2007, we have been passionate about designing and developing mobile apps— ones that are creative, intuitive, and so functional that they become indispensable to those who use them.

That's possible because we're committed to staying on the cutting edge of technology— not to be hip and trendy, but to enable our clients to achieve greater gains through the use of it. We believe a large portion of our success comes from being true partners with our clients, so we immerse ourselves in understanding what they need to achieve. We love to collaborate, and always seek to offer each other mutual respect and support.

Of course, working hard means we play hard too! Our location in Boulder is the perfect playground for all kinds of outdoor pursuits, and our team is regularly out and about. We are grateful for our community and seek to invest in it regularly through events like Ignite Boulder and Boulder Startup Week.

If you have an idea, or want to talk with us further about some of the concepts in this book, please contact us at info@inspiringapps.com or visit our website.

Trademarks

Apple, iPhone, Mac, iMac, iPad, iPod touch, iBeacon, Apple Watch, iCloud, watchOS, MacBooks, MacBook Pros, Apple App Store, Instruments, Xcode, Objective-C, Swift, and Safari are trademarks of Apple Inc., registered in the U.S. and other countries.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

BlackBerry and RIM are trademarks of Research In Motion Limited (RIM), registered in the U.S. and other countries.

Microsoft, Windows, Windows Phone, Internet Explorer, FoxPro and C# are registered trademarks of Microsoft Corporation in the United States and other countries.

Adobe, PhoneGap, Photoshop, and InDesign are trademarks or registered trademarks of Adobe Systems Incorporated in the U.S. and/or other countries.

Google, Google Play, Chrome, Android, Android OS, Android Lollipop, Android KitKat, Google Glass, Nest Learning Thermostat, and Google Analytics are registered trademarks of Google Inc. in the U.S. and other countries.

Amazon, Amazon Crash Reports, and Kindle Fire are registered trademarks of Amazon in the U.S. and other countries.

Barnes and Noble and Nook are registered trademarks of Barnes and Noble, Inc. in the U.S. and other countries.

Salesforce.com is a registered trademark of salesforce.com, inc. in the U.S. and other countries.

Appcelerator and Titanium are registered trademarks of Appcelerator, Inc. in the U.S. and other countries.

Rackspace is a trademark of Rackspace US, Inc.

Java and JavaScript are registered trademarks of Oracle and/or its affiliates.

Sencha Touch is a registered trademark of Sencha in the U.S. and/or other countries.

Ruby is a registered trademark of David Heinemeier Hansson.

Facebook is a registered trademark of Facebook in the U.S. and/or other countries.

Samsung and Samsung Galaxy are trademarks of Samsung in the U.S and/or other countries.

HTC and HTC One are trademarks in the U.S and/or other countries.

Starbucks is a registered trademark of Starbucks.

Xamarin is a trademark of Xamarin Inc. in the U.S. and other countries.

Kony is a trademark of Kony, Inc.

Corona Labs is a registered trademark of Corona Labs Inc.

Unity is a registered trademark of Unity Technologies.

4D is a registered trademark of 4D.

Dropbox is a trademark of Dropbox, Inc.

Evernote is a trademark of Evernote Corporation.

IBM Is a registered trademark of International Business Machines Corp.

Box is a registered trademark of Box, Inc.

QuickBooks is a registered trademark of Intuit Inc. or one of its subsidiaries, in the United States and other countries.

Bluetooth is a registered trademark of Bluetooth SIG, Inc.

FireChat is a trademark of Open Garden Inc.

Crittercism is a trademark of Crittercism, Inc.

Crashlytics is a trademark of Twitter, Inc.

Bugsense is a trademark of Splunk, Inc.

TestFlight is a registered trademark of TestFlight.

HockeyApp is a trademark of Bit Stadium GmbH.

Fitbit is a registered trademark of Fitbit, Inc.

Nike+ FuelBand is a registered trademark of NIKE, Inc.

Pebble is a trademark of Pebble Technology Corp.

Garmin is a trademark of Garmin Ltd.

Sensoree is a trademark of Neidlinger, Kristin.

Mood Sweater is a trademark of Sensoree

Ring Zero is a trademark of Logbar, Inc.

Siemens is a trademark of Siemens AG.

Mr. Coffee is a registered trademark of Sunbeam Products, Inc.

AT&T is a registered trademark of AT&T in the U.S. and other countries.

Sprint is a trademark of Sprint.

MobileIron is a trademark of MobileIron, Inc.

AirWatch by VMware is a registered trademark of VMware, Inc in the United States and other jurisdictions.

Flurry Analytics is a trademark of Yahoo! Inc.

eBay is a registered trademark of eBay Inc.