

"This one step - choosing a goal and sticking to it - changes everything." Scott Reed

A huge thank you to Valerie and Michael Holstein, my parents, for believing that a fifteen year old could start a company. You are the parents who said "yes."

Table of Contents

Introduction 4

Design 13

Refining Your Idea 14

Website 27

Idea Validation 34

Business Models 41

Fair Pricing 46

Planning Your Product 53

User Experience Design 63

Design Personality 81

Development 87

Funding Sources 88

Choosing a Development Method 100

Hiring Developers 110

Code Add-Ons 122

Beta-Testing 128

Deployment 135

Pre-Launch Marketing136

App Store Listing 137

Mailing List 144

Social Media 151

Online Community 159

App Video 171

The Press 178

Advertising Space 186

Discounts & Promotions 191

Introduction

Who This Book is Written For

This is a book written for people who aren't geeks. To read this book, you don't need to know how to program, do graphic design, or possess any especially technical skill sets at all. This is a book full of information that will help *anyone* make their own app, not just anyone with coding experience.

To get started, you should know the basics of how computers work, including how they store and transfer data, and the basics of how programs work. If you do not know these things, research the basics of how computer hardware and software works before returning to this book¹. Smartphones work the same as small computers do, although they have vastly different interfaces and uses.

If you don't like programming, you can still make your own app.

This book won't teach you how to code, because an app is more than code. Apps are small companies, and therefore require marketing, customer relations, product design, and all sorts of things in addition to programming. It is for this reason that expert programmers don't always create good apps, and good apps are not always created by expert programmers. Being a great programmer doesn't make you a great small business owner.

Programming the app is only one part of one of building an app; to place the entire book's emphasis on just that would be to neglect other important parts of a business.

If you plan and make intelligent decisions when it comes to your small app business, you will find success on the App Store. It would be a lie to say that this book has all of the "se-

¹ <http://www.dummies.com/how-to/content/how-does-a-computer-program-work.html>

crets to success” for the App Store, but you can count on success in the App Store if you work hard and plan. The App Store is not a lottery. Apps are not chosen at random for being featured, and it is not random whether or not your app becomes successful.

There is an element of luck and timing to success, but the elements of hard work and a good app play a much bigger part. You can not just whip up an app, throw it on the App Store, and watch the dollars roll in - nor will you slave away for years at an app, only for it to make no money. You will get what you deserve.

Do You *Really* Need an App?

Everyone these days has an idea for an app, from the garbage collector to the mailman to your boss. There is always an app for that, too, from a digital girlfriend app to an app that helps you manage multiple girlfriends.

It is popular to think that you or your business needs an app on the App Store. However, this might just not be true. An app is a fantastic platform, but an app is not an end in and of itself. Apps are just one way of getting things done on a computer.

To illustrate this, we will take a look at the relationship between desktop applications and websites in the early days of the internet - a good model of the relationship between apps and the internet today. Let us hearken back to 1987, the release of the color-display Macintosh II.



These were the days when everyone felt “you just had to be on the computer,” just as people feel “you just have to have an app” today. The Macintosh II computer had dedicated applications for everything, and you only went on the internet to access basic text and information. Heavy computing weight was pulled by standalone applications such as Word and Outlook, and the internet was just for finding information or for connections inside the dedicated applications.

This resembles the mobile computing experience now. You use your phone browser occasionally, but most of your time is spent inside an app. Apps are what cost your phone the most processing power, and the internet is utilized inside other apps more than it is by the browser.

But things did not stay that way for computers. As internet technology advanced, more and more functionality moved onto the browser. Dedicated games and applications have all but disappeared from your desktop, and all you’re left with are a couple of large and heavyweight applications such as text, photo and video editors, and in-depth games.

Most of what you do has moved onto the browser - you connect with your friends in the browser, your cloud storage is in your browser, your homework is done in the browser, you check your email in the browser... all things which used to be restricted to dedicated applications. What does this mean for apps on smartphones?

It means that things are likely going to go the same way for the smartphone.

Simple applications are going to move into the browser as better internet technologies become more popular and supported, and only big and heavyweight functions of smartphones are going to remain dedicated inside their own applications.

So no, apps are not the end-all-be-all of the smartphone. In fact, if the trend follows history, the app will be largely phased out. We're still at an early point in this evolution, where content-consuming apps remain as standalone applications on smartphones. Because of this, lots of apps do still belong as apps. But some common app ideas used to be appropriate as apps, but are not so anymore, like:

- Apps for a single website content-producing website
- Apps for a single ecommerce store
- Apps for a single band

What all of these ideas have in common is that they're ideas for single-content providers. Content aggregators make sense as standalone apps, as a one-stop-shop destination for the user. (This would be Amazon, Netflix, or the iTunes Store). However, single-producer websites are better off as responsive websites instead of apps within themselves, because the users aren't going to be visiting these websites very often. You don't want to force your user to download an app just to visit a website a couple of times a year.

That doesn't mean you should just forfeit the smartphone market, though. If your app seems like something the user wouldn't actually use all that often, consider if it's better off as a responsive website¹. A responsive website is a website which, works just as well on a phone as a native app would. Your idea is probably better off as a responsive website if:

- The user will not be visiting your app more than once every fortnight
- Your app is going to be an outlet for a single provider

¹ http://en.wikipedia.org/wiki/Responsive_web_design

- Your app will have a limited amount of content
- Your app will have static content (such as an encyclopedia)

If your idea fulfills these and you're still set on making an app, make sure you've thought through why making an app is better than just making your website work on phones. Your app must add so much value that users would prefer it over a mobile website, immediately.

Unnecessary Expenditures

You want to be on guard against unnecessary spending on your app-making journey. It can be tempting to spend a lot of money when you're trying to make an app, as you find things which will make you more legitimate. These can range from inexpensive things like URLs and business cards, to expensive things like web design services and consultation.

Don't let concerns about being official, legitimate, or anything else convince you to spend money where you don't otherwise need to. For the first version of an app, you're only looking to prove that the concept of your app works - and if it doesn't, you don't want to have lost a ton of money in the process.

Leave expenditures which make you seem legitimate until after your first version is released, until after you know you don't have a dud on your hands.

Do You Need an NDA

We're going to take a moment and specifically address this myth before we get started, because believing in it will seriously impede your progress throughout this book.

The myth is that you need an NDA to protect your app idea from Zuckerberg-style intellectual property theft. It's not true.

You do not need an NDA to protect your app idea.

NDA's *do* have their time and place. For instance, if you're Apple Inc. and are striking a deal with AT&T for exclusive service on the new-and-still-secret iPhone, you want an NDA. If AT&T were to leak that Apple was making a smartphone, it would have ruined Apple's plans and changed the smartphone market for years.

But, you are not Apple Inc. You are not striking a high-stakes deal with two billion dollar companies, which if revealed could ravage tens of thousands of families. If your idea is shared, you won't drop dozens of points in the New York Stock Exchange. You have nothing to lose, because you have nothing at all.

Furthermore, Apple had a real and concrete product, not an idea, when they were pursuing a deal with AT&T. The AT&T NDAs were not protecting Apple executive's *ideas* for an iPhone, they were protecting *the iPhone*. All you have is an idea.

An idea that, likely, someone has already had before. Probably many people have had before. When starting to make an app, all anyone has is an idea.

People Don't Respect NDAs in App Development

Quite apart from the needlessness of an NDA, it's counterproductive to even request one. Many industry professionals will outright refuse to sign one, and you don't want to lose the valuable resources of money and mentorship over an NDA.

One of the signs of a new app developer is an insistence upon an NDA. Whenever someone says "I have a great idea, but I can't talk about it without an NDA," those with experience don't think "ooh, I wonder how great their idea is! I'm so curious!" They think "Oh great, not another one." People who demand NDA's become repellant to the educated and successful when it comes to the field of mobile apps.

If you demand an NDA, your company probably won't get very large anyhow.

Lastly and most importantly, *the only way to validate your idea (make sure it's even worth the effort) is by talking to a lot of people*. You will have to tell people about your idea to do so. There is an *extremely thin* chance one of these people will steal your idea, but that segues into the next point.

It's all about execution anyways. Without execution, your idea is literally worth nothing. The only value an idea has is in the fact that it is being executed on; your app idea, if you decided not to make it a reality, would instantly become worthless unless someone else tried to make it a reality.

Ideas are so *worthless* on their own that Seth Godin put 999 of them on his blog for free¹. Look through those ideas and you'll notice that some of them are actually pretty great. If ideas were worth so much, Seth Godin would have sold them or protected them instead.

If you still feel that you ought request an NDA, make sure your reason for doing so is unique and compelling. Be able to defend your use of an NDA when you inevitably encounter resistance.

Affiliate Links

This is the disclosure policy about affiliate links in this book.

There are a lot of links in this book, and some of them are affiliate links. Indeed, there are affiliate links in a book you've already paid for. But the fact that links may or may not be affiliate did not influence whether or not they ended up in this book. Some of the links in this book are not affiliate, and some affiliate links did not end up in this book.

¹ <http://www.sixmonthmba.com/2009/02/999ideas.html>

What determines whether or not a link ended up in this book was whether or not the product or service referenced is good, not whether or not it had an affiliate program. If a great service had an affiliate program (which often benefits both referrer and referred), then that's even better.

Being open with readers is always better. Because of that, the footnote will always mention whether or not a link is an affiliate one. That means you can tell whether or not a link is an affiliate link by checking the footnote, and seeing if it is noted as an affiliate link and has an affiliate URL.

If you have a strong objection to participating in affiliate programs, go ahead and Google the link mentioned on your own to sign up. What we *don't* want is you using an inferior service because we placed an affiliate link you objected to.

If the recommended service was not awesome, there would be no link provided at all.

Design

“It is not simplicity on this side of complexity I am concerned with, but simplicity on the other side of complexity.” Albert Einstein

Refining Your Idea

A great idea with poor execution is a poor app, but a poor idea with excellent execution is a great app. This means that it matters far less what your app idea is, and far more that it's a really well made app.

This is true, but it's not the whole truth. Anyone who has failed at a startup can verify that no matter how well you execute a poor app idea, you can never be very successful, because part of what makes a poor app idea poor is that nobody wants to download the app. There is no *market demand* for the app.

So yes, having a good idea is key to success. Having a refined and thought-out idea is part of what makes an app a success. If you fly off the handle and start making the first thing that pops into your head, you'll have no idea whether or not it will sell. You need to assess and adjust your idea for success now, while it's still free and easy to do so. If you wait 'till later, it will be exceedingly more expensive to adjust your idea, like turning around on a road trip after you're already halfway to the destination.

This isn't just because you need a winning idea for a winning app; with over one million apps on the app store, your app idea is competing against hundreds of other apps based on *the exact same idea*. The humble task manager has thousands of iterations on the app store, all competing on execution and not idea. If you make a task manager with just a slightly better idea (e.g. group collaborating on tasks for school projects), you'd already have a huge leg up on the competition. This same crowdedness on the App Store is present in every category.

Contrary to popular opinion, you don't just magically 'have' a better idea. It doesn't happen in a brilliant flash of genius. Having a better idea involves taking a less good idea and making it better though hard work. This requires a significant time investment in developing your app idea. Analyze your idea by checking your current app idea against these idea refinement guidelines:

Make it Small in Scope

One easy thing to remember while refining your app idea is that it should be small in scope. This means that it should do only a handful of things, but it should do those handful of things really, really well. Think about all your favorite apps – Snapchat, the app that sends self destructing photos – Facebook, the app that lets you post messages, pictures, and comments to people or groups' walls (and that's a lot for one app – Twitter, the app that has 140 character messages – 2048, a game with exactly 4 controls where you slide tiles around– the examples are endless. These apps only do a few things, but they do those few things well.

iPhones are used in small bites, in moments and seconds, and their purpose should reflect this.

An iPhone game should resemble a flash game, not in graphics but in gameplay (quick play under 30 minutes, during class time or while the boss isn't looking). A productivity app should get you to the screen you need and get you out quickly. A utility app, even more quickly – most utility apps only have one or two screens at all.

Even the most in-depth apps, like document editing, won't be used for more than a half hour or so at a time. Users aren't powering through writing the great american novel on their iPhone. At best, they're capturing ideas on their iPad. Any long times spent in front of an iPad are done so during content consumption (watching tv shows and movies, or studying for class).

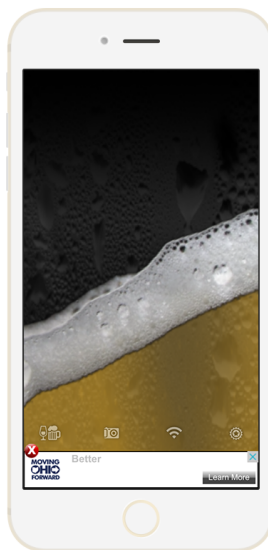
With so little time to make an impression, your app needs to make a good impression, and make it quickly. You can't do this with an app that has more functions than a swiss army knife.

Your app needs to be simple, new, and engaging, the first time.

Cool Features ≠ A Good App

Another easy guideline for app idea refinement is that you have to keep all features relevant to your app. Cool features do not make an app good.

When the iPhone first came out, using the accelerometer feature was very in vogue. But apps that became famous early, like iBeer¹, had ideas that used the accelerometer in an engaging and simple way.



Apps which did not become famous put the accelerometer feature before the app's purpose. One example is apps which would exchange business cards simply by bumping your

¹ <http://www.hottrix.com/iBeer>

phone or putting them near each other. These apps did not get nearly as big as anticipated, because it is a classic example of putting the feature before the purpose.

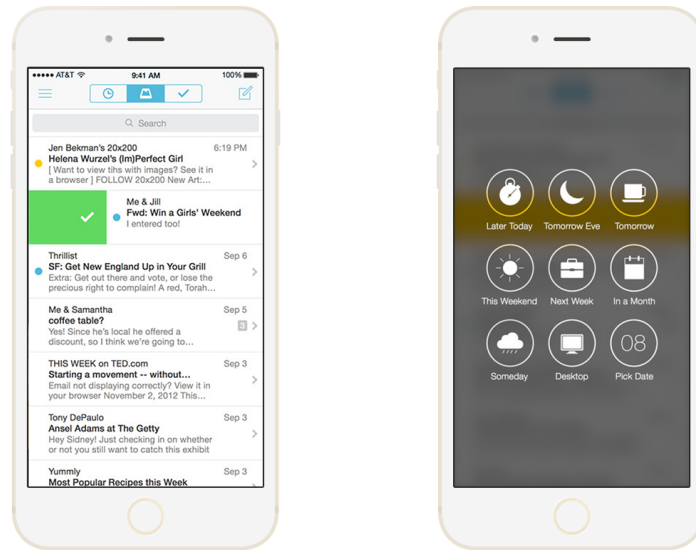
If you've thought of some feature that would be really cool on iOS, that's great – write it down and save it for an app it's appropriate for. While a feature can make an app cool, new, and different, a feature isn't a reason to make an app in the first place.

With new apps and cool features coming out every day, it can be easy to get wrapped into feature hype. But it's important that you're able to differentiate between a feature and a benefit.

Take the app Mailbox¹, an email inbox management app available for iOS and Android. It attracted a lot of media attention for having a huge waitlist, and subsequently being acquired by DropBox.

You might say "But Mailbox is cool because it has different features for eMail!" No, the idea of snoozing an alert or leaving it until later is not a new feature. Our smartphones have been capable of this for a long time. Mailbox is different because it applied these features to a new core concept, 'treating eMail like a to-do list.'

¹ <http://www.mailboxapp.com/>



As cool as some iOS features are, at the end of the day an app is purpose-driven. An app's feature has to do something in the context of the function of the app. If it doesn't help the user have a better and smoother experience, it should be left out.

The importance of this cannot be overemphasized. Differentiating on feature set won't win you the fight, but differentiating on core experience will.

An app has to have a good core function and a good experience to be successful in the market.

Apple MacBooks are a great example of this. On paper, they have less features and hardware than any other computers, and are astronomically more expensive. By every numerical measurement, they are a worse computer than almost any PC. Yet, they are top sellers because they provide a delightful experience.

Providing a delightful experience is what will win you the app market, and providing a delightful experience begins with a refined core app idea.

Compare With Your Competition

Every app has competition, even yours. Competition comes in two flavors:

- *Direct Competition:* Direct competition is competition which competes directly with you. They solve the same problem you do, in the same way you do.
 - For instance, Honda and Ford are in direct competition with each other in the automobile industry.
- *Indirect Competition:* Indirect competition is competition which solves the same problem you do, but not in the same way.
 - Taxi cab companies are indirect competitors of Honda and Ford, because all three companies solve the same problem (transportation of passengers), but taxi companies do so in a different way.

Your app may not have any direct competition, but it will always have indirect competition. This competition can be something as simple as a paper and pencil list, or something from another industry entirely. Whatever your indirect competition is, you have it.

What You're Really Saying

When you say you don't have any competition, you're really saying one of these things:

- "We have competitors and *don't know it*."

This is not inspiring. You simply cannot go into business without knowing how your product is better than your competitors, direct or indirect. Find your competition and pin down how you are better than them for the user.

- "We sell a substitution product."

For instance, you sell margarin and all your competitors are butter companies. That's fine, but you still must make sure it's obvious why someone should buy your margarin, and not their butter. Don't rely on people to figure out why they should buy your product on their own.

- “We have no competitors, because nobody wants an app which does this.”

If your app is unique and you have no competitors, that means your app probably doesn't solve a painful enough problem for your users. This circumstance is unlikely - what is far more likely is that you have competition who is not good at marketing, or you are in a niche category.

- “We have no competitors because we have something awesome people don't know *they want yet.*”

It cannot be overstressed how unlikely this is. And even if this is the case, you will have competitors almost the second your app launches. What's most likely is that your idea is not as unique as you think.

Finding Your Competition

If you don't know who your competition is, go out there and go find your competition.

You can start this search by answering a couple questions about your user:

- What are your users currently doing to overcome the problem they're having?
- If users didn't have your solution, what would they do?

Find your competition, and pin down why your app is better.

People who lack direct competition have a marketing challenge ahead of them; explaining to users why your substitution app is better than the app they're already using. It's easy to convince people to switch from a car to a better car, but not so easy to convince them to switch to a taxi.

On the other hand, in the war between two apps which directly compete, users will download the app that's been reviewed more and is more popular. So while it's easier to convince people why they need this solution to this problem, it's harder to convince them why they need *your* version of the solution. This is due only to the fact that your direct

competition is higher up on the charts and easier to find, because they were there before you.

Because other apps (of any kind) will be higher up on the charts and more reviewed, if there isn't a clear reason to choose your app, users won't. There needs to be a very real and very obvious reason why your app is better, a reason that isn't just tied to the ratings and chart position.

Furthermore, what is obviously better to you isn't obvious. If it's not a short sentence and immediately obvious after seeing *just one* screenshot of the app, it isn't obvious enough. This is all the more investigating a user might be doing, and you need to impress them in that brief amount of time.

Differentiating is quite the challenge, but the field is competitive with 1.3 million apps on the App Store. More apps are added every day, and the store is only getting more competitive.

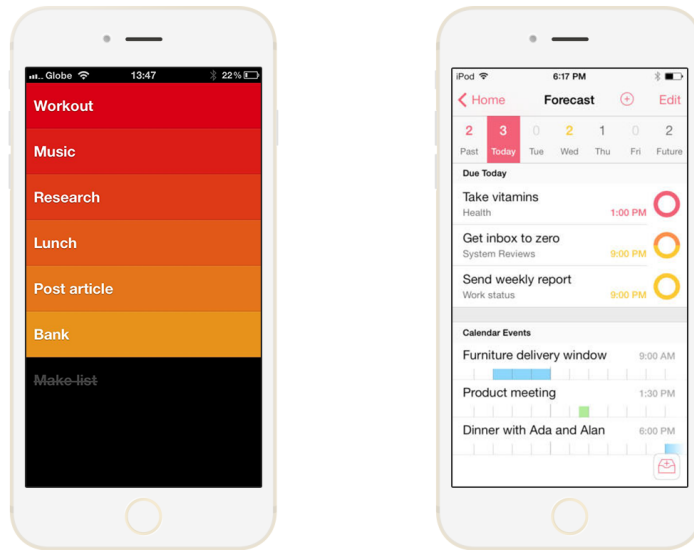
Execution is Everything

Execution is everything. You can create an app which makes your iPhone print money, but if they can't find the "print" button, folks won't use the app.

Whatever your app does, it has to do it simply, and has to do it well.

It doesn't matter how many features your app has, or how fancy it is. Consider Omnifocus¹; it is considered the biggest and most robust task management app on the App Store. It syncs with just about everything, reminds you of to-do's based on time and location, and you can attach links and videos within the app.

¹ <http://www.omnigroup.com/omnifocus/>



On the left, Clear. On the right, Omnifocus.

Plenty of people like Omnifocus, but just as many people like Clear¹. Clear is also a to-do app, and is very basic: Pull down to create item, swipe to complete, pinch to see lists and then expand again to see the items in that list. It doesn't have nearly as many features as Omnifocus, but has enjoyed a lot of success – about 2,000,000 downloads' worth.

This is evidence of the fact that a well-executed idea can lead to success on a bigger and grander scale than features, any day. Make sure that not only is your app idea small in scope, but that what it does do is executed in a way that is simpler than anything else available.

1 <http://realmacsoftware.com/clear>

Make It Easy to Explain

Another reason iBeer was so successful was because it was exceedingly easy to explain. - "It's like beer on your iPhone!" Temple Run¹ is similarly easy to explain - "You've got to avoid the monkeys and not fall off!" Or 2048², another game which went viral recently - "You've got to combine numbers to get to 2048!"

Things that are easily explained easily go viral³, because all it takes is one sentence exchanged from one friend to another for a download to happen. By contrast, anything that is difficult to explain will not go viral, no matter how great it is, because people won't invest the time necessary to understand.

A good way to measure how easy your app is to explain is by handing your iPhone app to someone, and then asking them to tell their friend about it. Don't provide them any prompting or help; you don't want to provide them with good ways to explain your app or good things to say, because you want to see what a random person out in the world would say to explain it.

If your app idea can't be explained in a sentence by someone else, you need to refine it further. Test different one-liners for your app over and over with people; once your app is easily explained, you're golden.

Always Get Feedback

One of the best ways to measure your app idea is to get feedback. Feedback is important throughout the whole app-making process because it is a reality check as to what your users want.

1 <http://www.imangistudios.com/index.html>

2 <http://ketchappstudio.com/>

3 <http://www.urbandictionary.com/define.php?term=Virality>

Getting feedback early and often is one of the keys to creating an app that people love.

Below are some ways you can get personal feedback on your idea.

Ask friends and family for feedback.

This is something you can do right now; stop reading this and type up an email to a close friend with your app idea, and ask them what they think of it. Make sure it is someone who is comfortable with providing you honest criticism, and won't just tell you it's good to encourage you. If your idea doesn't pass this initial test, modify it before moving on to other methods of validation.

Go to local events for your target audience.

These can be found through the website Meetup, especially in the tech section. Set the distance to your area, and attend some of the app and startup meetings. Tell people there about your app idea, and solicit feedback from them. Connect to people you meet on LinkedIn, and grow your personal network. A personal network is one of those things that you don't need until you need it – and then you really need it.

You've got your user feedback on your idea now— be careful not to put too much weight on their feedback.

Interpreting User Feedback

Knowing what your user wants and needs is a critical part in designing a successful app, but you need to be careful not to just mindlessly chase the features the user asks for. User feedback can help center your design, but you need to always keep the core purpose of your app in mind, the purpose you just spent time pinning down.

If you discover that what your users want is completely different than what you initially had in mind, and you want to make that different thing, it is known as a product

pivot¹. You can pivot your app if you want, but you have to throw out all the work you did for the old idea and start brand new in order to make a quality app. Don't hang on to old work just for the sake of having done the work. This is not like an hourly job; each hour spent is working towards something, and you're not counting the hours – you're counting distance towards your goal.

If you keep your app's core purpose in mind, sometimes it's appropriate to ignore user feedback. This is counterintuitive, but it's also a principle that Apple Inc. lives and dies by.

"A lot of times, people don't know what they want until you show it to them." Steve Jobs, BusinessWeek, May 25, 1998.

This is an important thing for you to keep in mind as well – the user accurately knows what problem they're having in their life, but they're not always most deft at identifying the appropriate solution.

If merely doing exactly what users say resulted in a good product, Apple would not be as successful a company as they are. Apple Inc. regularly ignores feature requests and has a limited scope of compatibility, which (next to price) is the most frequent complaint made about Apple's products. Yet, they still manage to be beloved and exclusively bought by millions of people, raking in billions of dollars.

So what feedback *does* Apple incorporate into their products? Apple look at what the users want not from a technical standpoint, but from a more emotional one. Apple asks themselves what features would make the users *feel* good, not what features will make their lives better.

This means you should ask the users how they currently *feel*, not what features they want. Often the best features have never crossed the user's mind, but revolutionizes their life.

¹ http://en.wikipedia.org/wiki/Lean_startup#Pivot

This is evident on Apple’s website: Go to the product page of any Apple product, and it won’t have a laundry list of features and comparisons to competitors – it will have beautiful pictures of their products, and emotion-evoking statements about them. Nothing more.

Focus on what the users say the problem is, but don’t necessarily implement the solutions they suggest.

Give the problem honest, concerted thought, and come up with a solution (a core purpose for your app, as we discussed earlier), that evokes buzzwords like “innovative” and “intuitive.” Find out how much your idea or current solutions are failing, but don’t get caught in the trap of obsessing over features.

The Final Test

The final test for app idea development is when someone will inevitably ask about your app. You must be able to answer these questions clearly, in under two sentences:

1. What is your app?
2. What does it do?
3. Why should I download it *over competitors’ apps*?

Until you can answer these questions, your idea isn’t good enough.

Keep refining your app idea, and keep soliciting feedback, until you can always clearly answer these questions in a way that fully satisfies the person asking.

When you tell your app idea to someone, they shouldn’t be able to ask “Why wouldn’t I just do this?” or “Wait, how does it work?” afterwards. They should understand the first time you explain it.

If you do get the same follow up questions over and over, hit the drawing board again. People should be able to figure out the answers without your help. Also, you don't want an app you continually have to explain to people over and over.

Don't forget that all of this feedback is so you can get people to pay for your app. So while feedback like "This is a great idea" is gratifying, it is not the same as a commitment to buy. If someone says "I would buy this," write down their contact info. Their feedback is a lot more valuable to you than feedback from someone who is not a potential user.

Sample End

This is the end of the sample.

If you are interested in continuing the book, please head back to the [Idea to App website](#) to purchase the book through the outlet of your choice.

Idea to App costs \$19, but after it's release the price will increase to \$25.