8 THINGS TO LEARN BEFORE

MAKING YOUR APP

DANN BERG

Table of Contents

Preface

When I decided to make an iPhone app, I went into the process completely blind. At the time, I was working in retail after earning a degree in English, and had no previous experience with programming or building products. As a result, I headed down a lot of unnecessary paths and wasted a lot of time before I was finally able to complete my first app, Reader Tracker.

I had learned an important lesson in the two years I struggled to get an app in the App Store: you don't need to learn how to code if you want to make an app. With this in mind, I've helped people from various backgrounds and with a wide range of experience turn their own app ideas into actual products.

Having a strong grasp on both your app idea and your approach to work is crucial for success. I've seen so many people waste countless hours doing the wrong work or fixating on a dead-end idea, when there are a few simple exercises that can help save time, money, and sanity.

In this book, I discuss what you need to know before you even start working on your app — giving specific examples to illustrate these principles. Whether you need help with a current project or are just getting started on something new, you can use this framework to help approach your work and develop your idea to avoid making many of the common entrepreneurial mistakes.

I've learned something interesting while teaching my class The Non-Programmer's Guide to Getting an App in the App Store: not everyone actually *wants* to make a successful app. So many people are stuck on their own idea that they're not willing to make any changes, even if it would turn a dead-end idea into something massively successful. Many of these people will ask for advice from the wrong people or, even worse, never talk about their idea at all. They think they want success, but they make assumptions instead of putting in the necessary work, and their apps eventually fade into oblivion.

Turning an idea into an app isn't easy, even if you decide to outsource the development. It's going to take a lot of work and some trial and error, and it's going to involve a drastic change in the way you think about technology.

Building a successful app starts with a well-formed idea. Mobile apps don't just solve tech problems, they solve life problems — so you need to understand the impact you want to have on people's lives before you even start.

The most important message I've learned by creating my own app and teaching The Non-Programmer's Guide to Getting an App in the App Store is that you don't have to be technically inclined in order to create a technology product. I built my app while working retail, and I've used that experience to coach other people in non-technical fields — such as lawyers, accountants, musicians, and artists — to do the same in their free time. All it takes to create a mobile app is the proper mental framework and a little guidance.

If you've really decided to make an app, it's important to make a tangible commitment to yourself. Pull out a piece of paper or open a new document on your computer right now and write down the reasons why you want to make an app. They can be:

- I want to create a source of passive income for myself
- This is an app I want to use and nothing else exists
- I want to be able to tell people I made an app
- I want to help people
- I have unique knowledge that I want to share with the world

Even if you know these things in your head, taking the time to write your goals down will increase your changes of success. Keep this list in a convenient place and review it on a regular basis. It's important to constantly re-read your "why" list because it shapes the way that you approach your work each day. It takes the focus off the individual tasks, which can be stressful or seem overwhelming, and grounds you on what you stand to achieve from success.

It's important to make a tangible commitment to yourself. Take the time to write your "why" list.

From the moment I decided to make an app until the day that Reader Tracker was available for download in the App Store, I felt like I had unfinished business. I knew in my gut that I was meant to create an app, but I was always just a few steps away. I started down a false path, learning to code, but was eventually able to figure out what I needed to do to reach my end goals. Reader Tracker’s release day was one of the best days I’ve experienced.

There’s a long road ahead before you’ll have your final product. But with the right mindset and framework, it’s an achievement within anyone’s reach.

1

No One Will Steal Your Idea

No one will steal your idea

In the class description for The Non-Programmer's Guide to Getting an App in the App Store, I specifically state that students should come to class with an idea and be prepared to discuss their app with other students. Every now and again, I'll have a student ask me for permission to workshop a fake app, rather than their real idea. When I ask his reasoning, the answer is almost always the same:

They don't want someone to steal their idea.

Keeping your idea to yourself will do way more harm than good, always.

It's common for "idea people" to fear anyone who is more technically capable. It's the classic story of the Winklevoss twins and Mark Zuckerberg: the technical wiz-kid is able to build a multi-million dollar company by stealing someone else's idea. The fame and glory is stolen from the hands of the true founders.

But that plot is missing a very important point: social networking wasn't a new idea. Myspace was huge at the time, and Friendster before that. Having an idea isn't valuable at all — it's the execution that really matters. An idea *always* changes throughout the process of execution.

First-time entrepreneurs greatly overestimate the value of an idea. It's the execution that really matters.

The thing that made Facebook a success was the execution, not the idea. There might as well have been a hundred people working on building a digital yearbook (the original vision for Facebook) — the victor will always be the one who can get the most users.

The desire to safeguard your idea by never letting it out of your brain is hurting you more than helping. You're refusing to let the idea grow and evolve into something that's actually useful. It's suffocating inside your head.

It's dangerous to make assumptions about the problem you're trying to solve, yet first-time entrepreneurs often go into hiding rather than trying to test their idea. They'll work in the dark for months or years, perfecting their personal vision of a product, only to realize that no one actually wants to use their app. By that point, the time is already lost and the money already wasted.

That's because first-time entrepreneurs greatly overestimate the value of an idea. If I see one more "idea-preneur" trying to find a technical cofounder by offering 50 percent equity (or less!), I'm going to scream. What value is being adding to the company while the developer is risking time and good money building the untested idea? Saying that that a button should be on the left instead of the right isn't value. It's just taking advantage of a developer (if the "idea" person can actually find one).

There's no such thing as a new idea. Even if you've never seen it before, there's a good chance someone else is building a product that's extremely similar to your idea. But the cool thing is that it doesn't matter. In fact, it may actually work for your advantage, as I'll discuss later.

More often than not, when people claim to be afraid someone will steal their idea, they're actually just making excuses for the real fear. In reality, they are afraid to talk about her idea and get real feedback because it makes them feel inadequate. They're afraid that talking about her idea would immediately put them on the defensive, with people asking questions they don't want to hear and making the idea sound lousy. But that's exactly how good ideas are formed.

There is no such thing as a new idea.

Feedback always makes you stronger.

You need to allow your idea to grow and change to meet your user's needs, and the only way to do that is to talk to mentors and potential users. The way you describe your app to others has a profound effect on the final product. Your elevator pitch needs to be polished and your idea needs to intelligently evolve before you even start design.

Anyone with the ability to execute on your idea already has hundreds of ideas of their own. They're not going to steal your idea because it would be number 548 on their project list, and they understand the amount of work it takes to execute even the most straightforward idea. These are the type of people you should be seeking out, not hiding from. They'll give the best and most honest feedback.

If you haven’t yet executed an idea, it’s scary to discuss your planned projects. But talking about your idea with everyone will only increase your chances of success.

2

Starting With An Idea Can Hurt You

Starting with an idea can hurt you

I had a student in one of my workshops who traveled all over the world for work, but always found herself dining in restaurants close to her hotel. She wanted to find exciting new places to eat in each city, but it was too much hassle trying to find restaurants and translate reviews after a long day of work.

She was taking my class to create a Yelp-like application, where users could rate and review restaurants around the world in order to build a quick-and-dirty dining guide for the frequent traveler.

The first thing I teach in my class is to figure out what problem you're solving and to separate that problem from the solution you're providing. In this particular instance, the student deduced the user's problem: "*I'm in a new city and I don't know where to eat.*"

If you have an idea, you're already making assumptions about your user. You need to test these assumptions before doing any more work.

Everything else — the international travel, the Yelp-like functionality, the dining guides — is *how* the student planned to address this problem.

There's a chance that this app might turn out to be the next big thing, but its success is based on certain unproven assumptions:

- Users will spend time writing reviews of foreign restaurants after one visit
- People eat near their hotel because they don't have access to a good dining guide
- People have internet access while abroad
- Users are willing to travel more for better food

When you enter a project with an idea already in your head, it can be a lot harder to adjust your thinking based on feedback. Having an idea means that you're already making assumptions about your user and his needs, and it's easy to immediately start building a product without first testing those assumptions.

It's important to separate your user's problem from your proposed solution.

It's important to separate the user's *problem* from your proposed *solution* (the app idea). If you already have your app planned out, you're making guesses about a solution that might not actually work.

Before wasting time and money, it's important to talk to your potential users about the problems your app hopes to solve. Are people willing to write reviews for your app? Is there really a lack of dining guides, or are people not even trying to find one? How are people currently getting restaurant recommendations while abroad?

It’s often best to enter a new app project with only a strong sense of the problem, and be open to the solutions proposed by your potential users. But time and time again, I see valuable feedback falling on deaf ears because the entrepreneur is convinced that a user’s behavior was wrong, not their idea. A user’s behavior is never wrong, because it’s what the user is actually doing.

Hone in on a single problem you’re trying to solve and find the solution by talking to potential users.

The best way to avoid this pitfall is to hone in on the single problem you’re trying to solve (such as: I don’t know what to eat; I don’t know what to do; I can’t remember what I’ve read) and find the solution by talking to potential users. Your app idea is just one solution to a problem, and there’s no way of telling if it’s the right solution until you’ve tested it.

For my globe-trotting student, the vision for her app was much more malleable by the end of the workshop. She proceeded to talk to friends and colleagues who also travel often for feedback on her idea. With focus on users who simply want a good place to eat abroad, she eliminated her reliance on user reviews and instead focused on translating existing online reviews.

With that small tweak, her app has a significantly higher chance for success.

Go into your app design process searching for answers, not providing them. Your final product will be much better.

3

You Don't Need to Learn to Code

You don’t need to learn to code

When I first decided that I wanted to make an app, I went out and bought a highly rated book on Objective-C programming (the language used for iOS development). I was able to understand the first chapter, but the second chapter made absolutely no sense. Thinking it might be that specific book, I went and bought another, only to have the same experience. Over the course of two years, I tried online video courses, tutorials, and even signed up for an NYU iPhone development course. While my understanding of programming improved slightly, I still didn’t have the skills to write my own app.

As I was struggling with learning to program, I was also reading a ton of books in my free time. I was in the midst of working my way through a Top 100 Books list, and needed a good way to keep track of the novels I finished. I found an app in the Apple App Store that suited my needs, but there were a few key features missing. I wanted to create a better version myself, and decided to build this app as a way to teach myself to code.

> **Many are stubborn in pursuit of the path they have chosen, few in pursuit of the goal.**
>
> — Friedrich Nietzsche

As I was planning out the app that would later become Reader Tracker, I realized that making an app isn't about code at all. Programming is just translating a fully-formed idea into computer language — that idea needs to be fleshed out and designed before even writing a line of code.

I decided to take a break from learning Objective-C and started working on the mockups and wireframes for my idea. Once those were complete, I hired a developer to turn my fully formed idea into code. It took ten hours of programming work for my freelancer, whereas it had already taken me more than two years to try and learn to do the same.

> **Learning to code isn't *always* a waste of time, but figure out your true goals before starting.**

Many people are absolutely terrified of building apps because they're afraid of the programming involved. But writing the code is only the last step in a long app design process. If you're doing the non-programming part right, your developer won't be making a single decision on his own. You should be able to hand over detailed documentation of your idea and get a fully functioning app in return.

I'm not saying that learning to code is a waste of time — thinking of an app and writing the code yourself is both challenging and rewarding. But it's important to understand your end-goal before picking up a book on Objective-C, or any other programming language. If your goal is to program an app yourself, then learning to code is obviously a step that can’t be skipped. But if you really just want to turn your idea into an app, you’ll save yourself a ton of time and effort by outsourcing the coding work. Time is our most valuable resource, so get a strong sense of your true goals before picking up your first programming book.

4

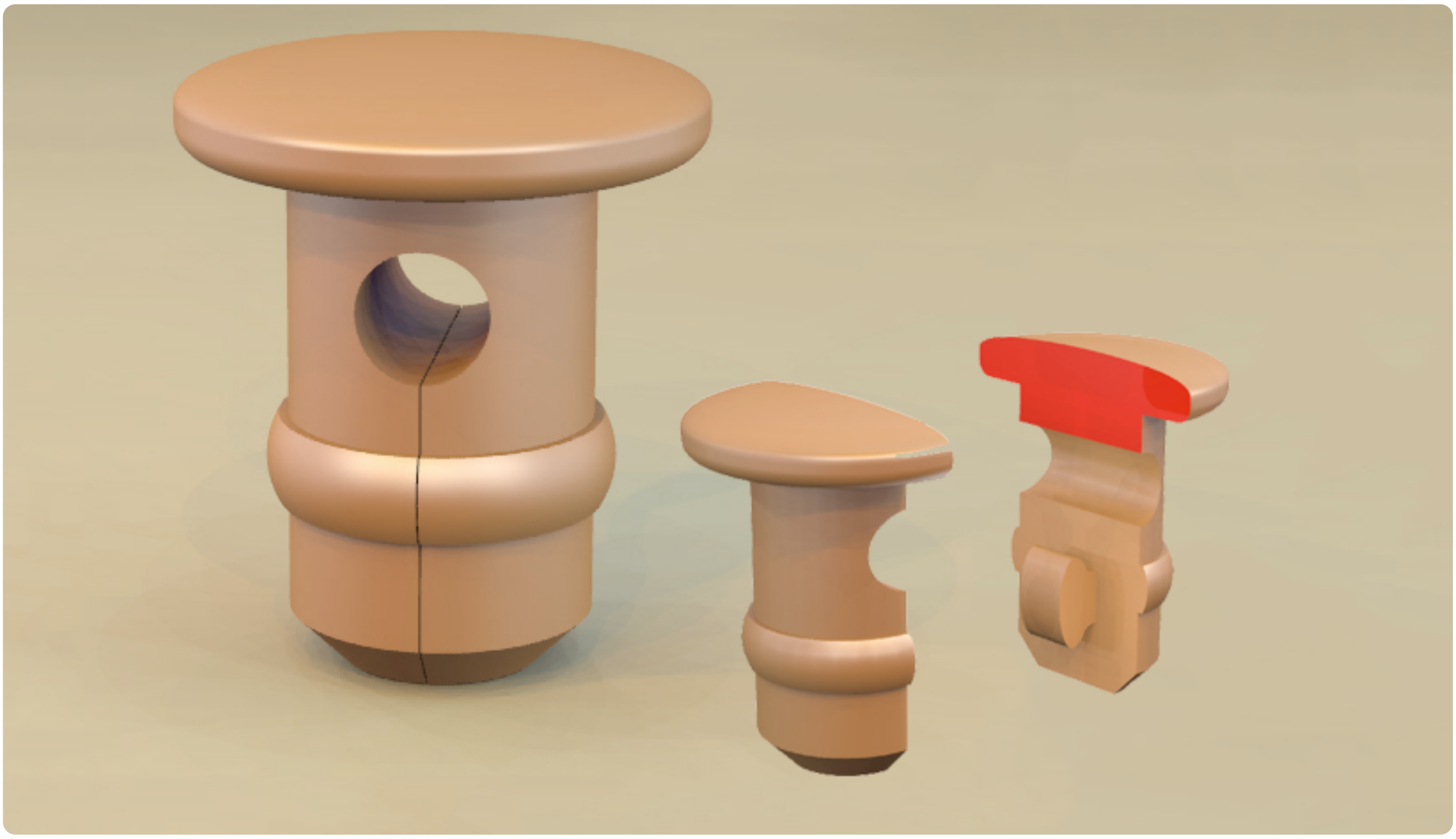
Not All Ideas Are Worth Your Time

Not all ideas are worth your time

Before I knew anything about building products, developing ideas, coding, or hiring developers, I read Tim Ferriss' *4-Hour Work Week*. In it, he discusses building a "muse," which is his term for a business that requires little work but generates significant income. The book outlines, step by step, how to create a self-sustaining business by finding a manufacturer, distributor, and customer service call center. Once you create a system where the distributor contacts the manufacturer when supplies are low, and customer service issues are neatly handled, you'll have a fairly hands-off business that exists to make you money. As the entrepreneur, all you need to do is check in on things every once in a while and work on marketing the product.

The entrepreneur just needs to put together the pieces.

Inspired by his methods, I came up with the idea for a product called Dust Plugs. At the time, my headphone jack on my



Don't be fooled by small victories and ignoring the warning signs that your idea won't work.

iPhone 3GS had been non-functioning for months — the result of layers upon layers of pocket lint. I envisioned the Dust Plug as a tiny piece of hard silicon designed to slip into the headphone jack and protect against dirt and dust.

But what really set the Dust Plug apart from the competition was a unique design that allowed the plug to snap onto a headphone cord when not in use. Thumbtack-sized pieces of plastic are easily lost, so creating a way to store it securely while listening to music was the key.

I bought the domain dustplugs.com and threw together a landing page with a big "buy" button, which prompted the customer to enter an email address instead of taking an order, since I had no product yet. I made an educated guess as to the final price, and advertised them as a little higher than that. After sharing my landing pages on a few websites and doing some search engine optimization, “orders” started rolling in. I was thrilled — I had found my muse.

I continued the process, hiring someone to create the CAD files and starting the search for a manufacturer. But as I started really hashing out the details and running the numbers, I realized that there was one vital part of Tim Ferriss’ instructions that I had ignored. And this omission began to grow into a bigger and bigger concern.

Even though I estimated that the margins of my product were relatively high, the total price of the product was still extremely low. Even if I sold them for $10, I’d need to consistently sell a large number of Dust Plugs to make a profit that matched my paycheck at the time. By Ferriss’ definition, muse products are niche and fairly high-priced, allowing entrepreneurs to sell even a modest amount and still make significant amounts of cash. It was becoming apparent that my idea was better suited among the SKUs of an established accessory manufacturer rather than as an entrepreneur’s single product. Not to mention the costs and risks associated with pursuing a utility patent, which would be necessary if I wanted to license the idea.

> **The only way to do great work is to love what you do. If you haven't found it yet, keep looking. Don't settle. As with all matters of the heart, you'll know when you find it.**
>
> — Steve Jobs

Everything was adding up to be a lot of work for a very limited payout — if any at all. Plus, as I worked on the business, it became obvious that this wasn’t even the type of work I really wanted to be doing. I didn’t want to be filling patents, dealing with manufacturers, and managing a physical product. My interests were more in building digital goods and services, and the challenges of creating a physical product were just boring to me.

It was a difficult decision — especially with an email list of interested people and CAD files ready and in hand — but I dropped the project. The idea for Dust Plugs might have been good, but it turned out to be not worth my time.

Dust Plugs wasn't the only idea that I've tabled over the years. I had an idea for a website that would connect longform bloggers, another that was a social network for tattoo artists and fans, and even spent time working on a luxury hotel ranking website. There are tons of reasons why each of these projects wasn't worth my time. There's a chance that what you're working on now won't be worth your time.

Don't let the time you've already spent on a project fool you into wasting more time.

Here are some valid reasons to pivot your idea or move on to a new one entirely:

- You haven't yet proven that the problem you're trying to solve exists
- You've found that the assumptions you've made about the problems are wrong
- No one is actually using your solution to their problems
- Your monetization strategy is proving unworkable
- You're not enjoying what you're doing

You may realize it's a dud now, or it could be next month, but you need to be prepared to stop and cut your losses if that moment ever arises. Once you realize that an idea isn't worth your time, you need to make an immediate change, either by pivoting to a new idea inspired by the same problem or moving on to something altogether new.

It can be difficult to completely drop a project that you've worked hard on, but if you know something is a flop, it doesn't deserve a single more second of your time.

5

Self Doubt Is Both Normal and Healthy

Self-doubt is both normal and healthy

Anyone making a product has spent a good number of nights laying in bed, unable to sleep, thinking they're building the *worst product ever* and that *no one could ever possibly want* what they're building.

These feelings aren't just relegated to late-night worries. You'll get a feeling in the pit of your stomach telling you that *all this work is for nothing* when you're both working on your wireframes or walking down the street. You'll tell yourself, "I should just give this up," over and over again. You may not even realize it, but deep down something might be telling you that you don't really deserve to make an app.

DON'T LISTEN.

Having feelings of self-doubt are more a sign that you're on the right track than proof that you're moving in the wrong direction. If you’ve never built a product before, the process of creating something out of nothing will be a foreign concept to your mind. Your brain may be trying to sabotage yourself to maintain the identity of someone who does not build products, so it acts up when you’re on the path to success.

As long as you have a strong sense of the problem that you’re trying to solve, and you have proof that the problem actually exists for a variety of users, then you’re on the right path. Don’t let anything tell you otherwise.

> **Let me realize that my past failures at follow-through are no indication of my future performance, they're just healthy little fires that are gonna warm up my ass.**
>
> — Ze Frank

Don’t be afraid to ask for help. Email me, write to the developers of your favorite apps, and post questions in online forums. Everyone who has ever created something has been in your position before, and most people are more than willing to help people out or lend a word of advice. You don’t have to go it alone. The more you talk to people, the more you’ll learn and the better your product will be.

When you realize that doubt and nervousness are signs that you’re on the right path, you’ll be able to plow through the tough times.

6

The More Competition, the Better

The more competition, the better

In early 2012, tech blogs and app websites all started talking about a new to-do list application called Clear. This was by no means the first to-do app in the App Store — there were already thousands at the time. Not only that, Clear cost 99 cents, while much of the competition was free.

A quick Google search will almost always reveal more competition than you thought.

This raises some questions, like:

- Why on earth would Realmac Software spend the time, money, and resources to release a to-do list app into an overly saturated market?
- How did they get away with charging money when there were tons of free options?
- Why was Clear such a success?

Before I answer these questions, I want to tell you about one of my students who made her living as a nutritionist. She wanted to create an application that would help people with dietary sensitivities figure out what they can and cannot eat while dining at different restaurants.

As soon as she told her idea to the class, a student said that her idea sounded familiar. Sure enough, a quick App Store search revealed a few different applications offering similar features. There was one with a database of safe foods, another that showed safe menu items for popular restaurants, and another that allowed users to scan a barcode to instantly check if a food was allowed on a specific diet.

The more solutions, the bigger the problem.

The student looked dejected. She was ready to give up her idea, but **was that the best thing to do?**

There are so many ideas that die after a quick Google search. It invariably turns out that someone has already built the product we were thinking of, so we decide it's time to wait for the next hit of inspiration.

This thinking isn't just wrong — it could scare you away from creating a killer app.

When there are a lot of products that perform a similar function, it doesn't mean that your life will be harder if you make another. On the contrary, it proves that there's a market for your app and it will be *easier* to find users. The more apps you find, the larger the market.

The important part is how you differentiate your app from the existing solutions. Your application may offer a feature that's slightly different, or have a unique design or interface.

If you haven't heard of the existing solutions, you can bet that there are a ton of people who haven't heard of them either. If you're craving a solution, there are other people who are also feeling the same way. Find these people, and you're all set.

Returning back to the Clear to-do app, Realmac Software created a unique solution to a problem that has always existed and will always exist: I can't keep track of everything I need to do. Clear has dead-simple features and a beautiful and unique gesture-based interface. It provides an elegant solution even though other solutions already existed at its release, and people responded by handing over their money.

Don't take competition as a sign that you need a new idea — think of it as free market research. Hone in on the problem and make your own elegant solution.

7

Making an App is the Easy Part

Making an app is the easy part

It may seem like making an app and uploading it to the App Store is the bulk of the work, but really it's just the beginning. Once your app is available in the App Store, it becomes one of over a million available to the potential user.

Making the app is the easy part — getting people to download your app is the real work.

There is a possibility that attracting users might not be a big concern for you. You don't need to think about marketing if:

1. You're creating the app as a hobby
2. The app is for existing users of another system of yours
3. Your app is an internal business tool
4. You're famous and people will buy anything you touch

If none of the above apply, you should start thinking about monetization and marketing strategy **now.**

Making the app is the easy part — getting people to download your app is the real work.

There are a few different ways to monetize an app. The first is to charge a fee for your application. Reader Tracker costs 99 cents, which means that I pocket 70 cents per sale after Apple takes its 30 percent cut. This is a straightforward model: if your app provides immediate value, charge your users for your app.

Another method is to make your app free and use ads to make money. Apple has an ad system that's easy for developers to incorporate into your app. If you think that you'll be able to attract a large number of users, but your user base may be hesitant to pay upfront, then using ads may be the best way to make money from your application.

A popular monetization option at the moment is in-app purchases, although this can be a little tricky with common utilities. With this model, an app is available for free in the app store, but users need to pay a fee to unlock certain features. This is great because users are able to try your application first to see if they find it useful, and if you've put enough value behind the in-app purchase, your conversion rate will be a lot higher than simply charging upfront for the entire app. Figuring out how to apply the in-app purchase model to your idea can be challenging, depending on your app.

Blog coverage is important for attracting attention to your app, but usually results in a spike of downloads rather than sustained traffic. Still, these can be a powerful tool if done correctly. If you've chosen to charge for your app, you'll get 100 free promo codes for every version you release in the App Store. Use these codes as gifts to bloggers. Never expect a review or mention, but always make sure they're aware of your app's value and there is a possibility they'll want to share it with their readers.

Always provide a way for the user to contact you.

Always provide a way for the user to contact you with feedback and questions. You can build this directly into your app with a help section, or create a nice landing page with the help of an inexpensive ThemeForest theme. The key to good copywriting is focusing on what the user gains from using your app (the problem that's being solved) rather than what your app does. It's a subtle difference, but vitally important.

I've only scratched the surface of marketing, and each tactic has various tricks and nuances that can optimize the effectiveness. Marketing is too broad a topic to broach in detail here, but it's something you should be thinking about throughout the app-building process.

It can take a long time to recoup the costs of making an app, but it's also a fun and rewarding journey that will teach you a ton of valuable new skills. Just keep in mind that making the app is only the first leg of the journey.

8

You Were Ready to do This Yesterday

You were ready to do this yesterday

Sometimes it’s really hard to stay focused. You’ll just be staring at a blank piece of paper, knowing that there’s a mockup to finish or email that needs to be sent, but suddenly you’re browsing Hacker News for articles about apps, browsing through your Read It Later list, or checking out that recommended book you had forgotten about. You will invariably find some article or post about the exact thing you’re working on, or something you know you’ll need in the future, and get sucked down a rabbit hole of helpful advice.

There is so much valuable information on the internet that it can be crippling. We’re always just one article away from having the skills we want — just one more blog post until we really understand what to do.

That’s bullshit.

When I graduated college and got a job selling suits at Hugo Boss, I had some experience in retail but knew very little about menswear. The first suit I ever owned was my work uniform, and I couldn’t even tell the difference between wool and cotton, or the benefits of each. There was a massive online training course that I was required to complete within two weeks of my start date, and I felt overwhelmed by the sheer amount of knowledge I’d need to learn.

Yet, on the second day of work, I had a customer who needed to get a new suit.

I was absolutely terrified during that first sale, and felt like I was doing everything wrong. I knew that every other sales associate had way more menswear knowledge than me, and there was a little voice inside my head that kept saying I needed to stop the sale and do some studying before coming back.

You learn more from doing than from reading about doing, even if you’re doing it wrong.

But 30 minutes later, the man handed me his credit card and I rang him up for a new suit.

Did I really need to have finished the entire suit-selling course by that second day of work to make that sale? Of course not. I just did the work. I listened to what a customer needed, helped him try on one of our suits, and ran his credit card when he said he’d take it.

As I went through the training over the subsequent weeks, I completed the training and picked up a ton of new knowledge, which helped me become a much better salesperson. After a few months, I could tell a customer’s suit size just by looking at him.

All that training and studying helped me become a better sales associate, but doesn't change one simple fact: **I made the same commission from selling that first suit as I did selling my last suit.**

I learned more from actually being on the sales floor and talking to customers than I ever did reading about suits. Sure, the online course helped, but it was more like icing on the cake than essential knowledge for selling a suit.

Reading news and blog posts is exactly the same. You may feel like you're learning new skills and being productive, but you're not actually getting anything done. None of the information matters unless you put it to use, and you don't actually need much information to actually start working. Even this book and my blog are meaningless unless you spend time actually doing the work.

The students that I've taught have always been *more* capable than they think, never less. If you think you're still lacking some required skills, you're just making excuses.

It's not skills that have held you back, it's mindset. By shifting your thinking about building products slightly, it can be the kick in the ass that you need to start **Getting Things Done**. Go out and start talking to your potential users, start making those sketches.

The fear of doing something wrong can be paralyzing, but the best thing you can do is create as much as you can as fast as you can — you can go over everything again later and it's much easier to correct a mistake than it is to make something out of nothing. Plus, you'll learn much more through the process.

If you're on the road towards building an app, you'll feel an indescribable sense of happiness once your idea is available in the App Store. Finishing a big project is nothing but completing a series of small steps.

You don't need to learn any more before taking action. People with way less knowledge than you are successful everyday. You were ready to do this yesterday.

The Next Steps

Get the program that will take you all the way.

This book is just the very tip of the iceberg — I haven't even started covering the specific steps and tactics I used to make my first app. If you've found this book helpful, I encourage you to check out the full online course, which is coming soon.

You don't have to, of course. In fact, I'd be just as happy if you took this book as a stepping stone on the path towards building an app yourself.

But if you do want a little more guidance, here's what's included in the online course:

- Worksheets for discovering the real problem you're trying to solve
- Specific methods for extracting valuable information from potential users
- Project milestones to keep yourself on schedule and motivated
- My exact scripts for finding, vetting, and hiring freelance developers
- Example PDF mockup files
- Techniques for monetizing your app

Click to get notified about early access to the online course.