

Before You Build

Everything you need to know before you start building your mobile app: How to validate your concept, how to get initial feedback, and how to turn your brilliant idea into a real app.



by Austin Betzer

Austin Betzer's

Before You Build

All rights reserved. In accordance with the U.S. Copyright Act of 1976, the scanning, uploading, and electronic sharing of any part of this book without the permission of the publisher constitutes unlawful piracy and theft of the author's intellectual property. If you would like to use the material in this book (other than for review purposes), prior written permission must be obtained by contacting the publisher at hello@strides.dev. Thank you for your support of the author's rights.

Published by Strides Development, LLC

strides.dev

Published in the United States of America

First Trade Edition: January 2021

Hey, thanks for being here!

Quick intro before we dive in. Hi, I'm Austin. I'm an app developer and I specialize in working with startups and founders and helping them turn their ideas into real apps on the app store. I recently took the leap from freelancer to "business owner" and created an app development agency called Strides and we've worked on 25+ projects and managed over \$1,000,000 in development budgets. We help people like you take big STRIDES and help them execute. (Get it?)

Introduction

Is there an idea you've been dying to turn into an app?

Chances are that we've never met, but if you're reading this now I think it's safe to say that you have an idea for an app and you're wondering if you should take the plunge.

If you're anything like me, before you start sailing through uncharted waters you want to have some idea of what success looks like.

There's a burning question you probably have that I want to help you answer and get some clarity around...

“What does it take to make my idea WILDLY successful? How can I see enough success to make the investment of money, time, and resources worth it?”

If you’re mulling over your latest and greatest idea and you have this same burning question, I wrote this for you.

What makes a successful app?

I’ve been on a LOT of app development teams and I’ve seen a lot of ideas. I’ve really been at each stage of the process — I’ve been a junior developer on gigantic projects, I’ve played around with launching my own app ideas, and I’ve been in the consultant role advising founders on how to launch their thing.

I’ve seen stuff work and I’ve seen stuff... well, NOT work. The difference? Have you ever heard the saying, “Bet on the jockey, not the horse?”

In short, YOU make a successful app. YOU are a bigger factor in your app being successful than your idea, your market, your industry, or your connections.

If you have that burning desire to make your app idea a reality, I want to give you a bit of guidance and point you in the right direction.

Let's put all that excitement, passion, perseverance and grit to work in the right direction. And as you go through this journey and you take the winding path, if we can be of assistance in helping you build your idea, we'd love to help you out.

Here's what we're going to cover in this eBook:

- **In section 1** we'll talk about validating your idea and how you want to approach nurturing your idea to turn it into an actionable plan.
- **In section 2** we'll talk about what it looks like to build your app — we'll talk everything you want to do PRE-CODE, we'll talk about what I think is the MOST IMPORTANT part of the entire process, and then we'll talk about what happens when you're writing code.
- **In section 3** we'll talk about launching your app and getting your initial user base and your first customers.
- **In section 4** we'll talk about putting your team together and making it all happen.

Section 1: Nurturing your idea into an actionable plan.

The most successful apps were successful IDEAS before they were ever an app.

I'm not talking about the *"damn that's a great idea for an app"* ideas...

I'm talking about the founders and thinkers who don't just have an idea about their APP, they have thought through how the app will be used, what success looks like, and what the real problem is that they are trying to solve.

The most successful products my team and I have worked on all had ideas that were thoroughly documented and validated.

Based on what I've seen work, this is my go-to advice when shaping your app idea. I'm gonna help you check yourself before you wreck yourself.

I am sharing this valuable information for slightly selfish reasons. I don't want you to get stuck in common pitfalls because if your idea rules, I'd love to see you come through my door at Strides.

Thinking about your app in these terms can help you save money, time, and effort. Wouldn't it be nice to avoid common issues and hiccups on your road to app store domination?

Ok, then keep the following in mind:

Start with things that don't scale.

What does "scale" mean in the app development world?

It means making the application serve more users. Scalability is the capability of your app to manage an increasing number of customers, clients, and/or users.

As you shop around for developers, you're going to hear this term a lot.

Developers will tell you *"this doesn't scale."*

You'll hear over and over again (but now at least you'll know what it means and why the developer is stressing).

Most developers will ONLY tell you to focus on ideas that are massively scalable — to focus on ideas that will solve a problem for MILLIONS of people.

Here's my advice. Start WAY smaller and be obsessed with things that don't scale.

Here's what I mean by that. If you're thinking a feature needs to scale to a billion users and you don't even have a live app yet, you're thinking way too big for this stage.

Here's a better approach: Think about how you can instantly bring value to the first 200 to 500 people.

Founders with overzealous and unrealistic scalability goals often miss the mark. They get more focused on vanity metrics like "total downloads" and they forget to obsess over what really matters... like how many people are using your app frequently.

Case Study: Air BnB



In the book “How I Built This: The Unexpected Paths to Success From the World’s Most Inspiring Entrepreneurs” Guy Raz talks about the early beginnings of Airbnb.

Everyone who heard their early idea hated it... “...the potential market opportunity did not seem large enough for our required model...” was one of the rejection emails they received from a potential investor.

Everyone hated it EXCEPT their customers. Their customers loved the idea.

In the book, Guy talks about how they would head out to New York over a weekend and talk to their customers ONE AT A TIME.

One-by-one, they would take professional photos of their places, update their listings to make them better looking on the site, and then they would usually get invited to stay for lunch or dinner and talk about how they could make the app better from the user's perspective.

THIS kind of user research is what made them wildly successful... but a lot of founders aren't willing to do something this hands on because it's "inefficient to talk to people one at a time..."

Do the work!

Have clear expectations and goals.

In the ideation phase, it's critical to develop realistic goals for your project. By getting crystal clear on the overall goals and expectations, you stay on track and stay effective.

App development is an overwhelming endeavor if you don't have a clear vision and a strategy.

(It's also really annoying to your developer when they ask, "How many downloads are you shooting for" and you say, "MILLIONS." Developers like concrete goals — "We are aiming for 20,000 app downloads in the first year with 4,000 people activating their trial.")

One of the best side effects of goal setting and reviewing expectations is it helps you cultivate a realistic mindset. It makes it that much easier to prioritize your money, time, and effort when you're clearheaded.

You'll be empowered to plan and spend with purpose.

You can avoid making decisions that will dilute your idea by doing your homework first. Focus on the basics: goals and expectations.

Focus on answering the following questions:

- What problem or pain points your app will solve?
- What is the purpose of your app?
- Who is your target audience?
- What authors does my target audience look up to? Where do they hang out online? How can I get in front of them easily?
- How many people will use the app in the coming 6-12 months?
- Are there goals related to monetizing the app?
- What are your current resources — financially and technically speaking? Where do you need more support?

- And finally, after answering all of the questions above, what's your elevator pitch?

From my experience, you can tell someone has really thought about their idea if they can pitch it quickly:

"I have personally been using ___ app and I've noticed that it's missing this simple feature that would make my life better. I asked 25 friends if they would use an app that solved that same problem and I had SUPER passionate responses. So I emailed my newsletter with a survey and I think this is a real pain point... If we just put XYZ feature together I think we could make a group of people happy enough to pay us \$X for the app!"

Or,

"I'm building an app in this industry. The target market is stay at home moms, an estimated 579 million people. We hope to catch 10% of the population. With a monthly subscription of \$5.99, we'll make X amount by 2022."

Two very different pictures right? One is very problem-focused and one is very opportunity-focused but they are both clear and specific.

Top development teams (like mine) only want to work with ideators who have clear vision.

If you can't clearly communicate your goals and expectations, you are not ready to create and collaborate.

Remember, clarity strengthens strategy.

Build an app for your customers, not yourself.

I'd consider this particular conversation a pet peeve of mine as a developer.

The conversation sounds like this, "Hey, I was using the app the other day and I don't like this feature. Let's redesign the entire feature to X."

If you want to be the creator of an app, you have to get your ego out of the way. It's no longer about YOU, it's about serving your users.

I always respond with a deliberate question when this comes up.

"Is this just your idea or have you ran this past your users?"

I know this is a tough pill to swallow as an innovator. Typically, we are inspired to create apps because we're solving a problem we've experienced ourselves.

But here's the thing, once that app is out in the world it's simply not just about you anymore. It's an offering to your users. And now your app's features and functionalities are on their collective plate — your personal opinion is just one of many opinions and doesn't matter as much as you think it does.

Don't send it back to the chef without a good reason — *"I surveyed 300 of our most active customers and they all got stuck at this part of the process."*

AH. Ok got it. Yeah let's fix it.

Make sure your app idea solves a REAL problem.

For an app to be successful and accessible it needs to provide consistent and instant value.

It all boils down to one truth, if this is not something that is not going to bring value to me every single day, I am not going to have it on my phone.

Real problems that drive people to seek at hand solutions occur weekly or daily.

An app that taps into a weekly routine or real world everyday problem is instantly valuable.

A prime example of this is the Day One journaling app. Day One is one of the biggest apps in Utah.

People love to journal but a lot of times they don't have their journal handy, or don't want to physically hand write a journal entry.

Journaling can be inaccessible and inconvenient. But that's where Day One swoops in. By offering a secure mobile process with templates, prompts, and notifications they've made it insanely easy.

Are you sold just reading that? Yeah me too, it's a really good idea.

Don't try and reinvent the wheel.

A lot of people want to have a very unique product, one that outshines the rest. I understand that, it's your little baby and you want it to be different and cooler than everyone else's.

Here's the issue with this, if you are building something completely different than ALL the other apps out there, nobody is going to know how to use your product.

You still want an app that is recognizable. It must be somewhat familiar to be marketable and user friendly.

Take your web browser for example — You may prefer chrome to safari but isn't the navigation basically the same.

Users don't want to deal with a considerable learning curve, or features that aren't intuitive.

You can avoid this pitfall by doing your research and keeping it simple when it comes to features.

Go into other apps offering a similar solution and review the app store's most popular apps.

Take solid notes. Take lots of screenshots. USE the apps. Pay for the premium features. Pull concepts, aesthetics, functionalities, and features from big-time products so that way your user base is already accustomed to your app when they download.

If you're building a feature that's never been done before, you're probably focused on the wrong thing.

A lot of the time, apps can be over-engineered which really means they're overcomplicated.

I actually experienced this recently with an app I am working on. The client had spent his whole life on this one app, diving into code and every angle of its design, and function. His app was clearly his passion, which I can 100% respect, but it had taken on a life of its own. And for whatever reason, things had gotten overcomplicated and out of control.

Everybody was trying too hard to do something groundbreaking and it actually just led this app to never get OFF the ground.

I didn't understand what these engineers were thinking. It was obvious they could have simplified code, and saved a ton of money and time in the process.

It also would have made it easier for future developers like me to work on their app and pick up where they left off.

Focus on your idea and the problem it's solving and try to think of the SIMPLEST way that feature could be built.

Most apps that hit the top charts aren't doing completely new things... they are doing the same, familiar things in a completely new way.

So here's how you do that:

Find things that are working for other apps and think of a unique use case behind those features.

What is a unique use case? It's essentially a description of how a person uses a process or system to accomplish a goal.

Look for REALLY GOOD features in other apps and think about how you could use that same feature to achieve a DIFFERENT goal and offer a different benefit to your users.

Here's an easy example:

	Lyft	My App
What does the app do?	Users can request a taxi right from their phone and request someone to come to their location and pick them up.	Customers can schedule a pool cleaning from their phone without having to call anyone.
Feature	Ability to see how far away your driver is after you've been matched with someone.	Ability to see how far away your pool tech is on the day they have been assigned to your house.
Benefit to User	User knows how many minutes they have until their ride is at their front door.	User knows when the tech will arrive! No more waiting around for an appointment between "8am and 3pm."

See what I mean?

Just by using other apps you can start to put a lot more detail behind how certain features work.

- “What if I need to cancel my appointment?”
- “Should I charge my customers if the pool tech is at their house for 20 minutes before they answer the door?”
- “What should the notification say when the tech is a few minutes away?”

Part 1 Wrap Up:

Launching a seriously successful app doesn't happen by accident. Every time I've met a founder of a huge app, they really know their stuff.

Do the work. Do the research. Take notes. It really makes a difference.

Section 2: Turn your idea into a real app.

I'm going to share the build process in **THREE PARTS**:

1. **First**, let's cover everything you'll need to do before you start writing a single line of code.
2. **Second**, I'm going to talk about "crossing the rubicon" by getting all of your rad ideas packaged up in a "CLICKABLE PROTOTYPE!" This is one of the least-talked about (and in my opinion most important) parts of building your app.
3. **Last**, let's cover what it looks like when you start actually "committing code" as they say...

Sound good?

Ok so let's talk about getting your idea down on paper.

Start with a giant feature list.

This is something you can start working on right away on your own — even if you are the least technical person on the planet.

A feature list is a giant collection of all the things you want to have in your app.

Let's say you want to build a meditation app. Open up a blank document and start cataloging everything you want it to do:

Things I want users on my app to be able to do:

- *Choose between what type of meditation they want to do*
- *Set a reminder and alarm that tells them it's time to start their evening meditation.*
- *Choose what they are feeling right now and find a meditation that will match that feeling.*
- *Make a note about how they feel at the end of their session*
- *Upload their own audio track from a voice memo*

When we do this with our clients we literally just start by thinking through EVERYTHING the app could possibly do. Then we start grouping the ideas together into categories.

This process will ultimately turn into your roadmap for rolling out your entire project.

Get an expert to write up a “technical recommendation” based on your features.

Once you have your features down on paper and you’ve started organizing them, you need to start thinking about what services you will want to use, and what languages you’ll use for your app.

There’s a lot to talk about here... WAY more than we can fit in this book. But here are a few things to keep in mind.

One thing to consider is if you should build your application natively or not. A lot of people think that building something that supports both iOS & Android you will save time and money. This is not always the case. If your application performs complex operations like video & image compression, peer to peer connections, messaging operations, ride sharing, streaming services, or has complex designs and animations you should think twice.

This means that you're not just going to build ONE app... if you want your app to be on iPhone, Android, and the computer, you are going to need a native app for each platform.

You'll definitely be able to reuse your strategy and a lot of your design assets, but as far as the code goes if you want to do it right you will need to build a custom app for iOS, Android, and Desktop.

Also think about your team size — if you were to do something that supports cross platform abilities odds are you will have a larger team to maintain the application as a whole. The smaller the team the smoother things usually tend to go in my experience.

At Strides, we write most of our applications natively but occasionally we will use React-Native or Flutter.

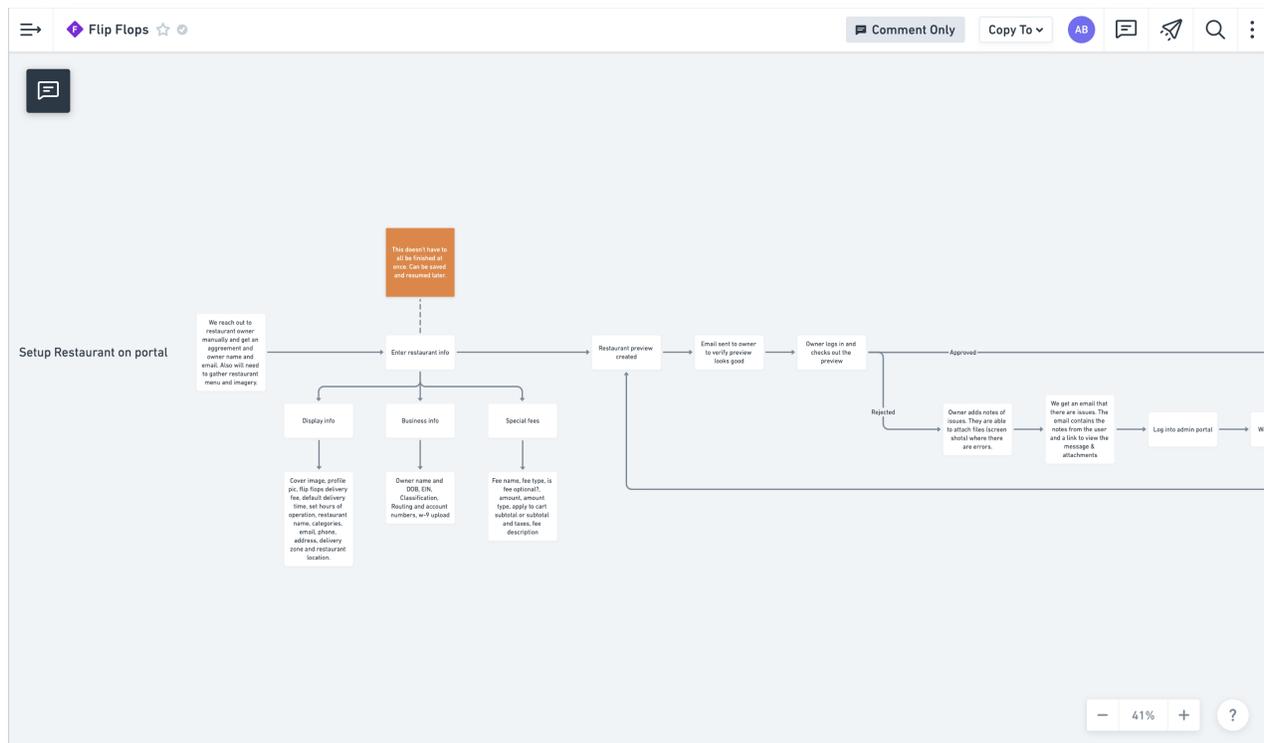
- iOS apps are written in Swift and Objective C.
- Android apps are written in Kotlin and Java

And then you'll have a backend database structure that ties everything together and this might be on Amazon Web Services (AWS) or Google Firebase.

Document every step your users will take with a User Journey.

In this step of the process we are mapping out the flow and functionality of your entire app without any design or code.

We recently built a MASSIVE food delivery app called FlipFlops and this is what their user journey looks like:



This document details all of the steps that a user can go through and everything that happens. If you have a settings page with 7 different buttons, each one of those buttons will be mapped to a page with it's own set of features.

With FlipFlops, we outlined EVERYTHING that happens when a customer places an order.

- What options do they have when they are selecting food from the menu?
- How can they customize their order?
- What happens after they enter their payment information?
- What does the employee at the restaurant see when they receive a new order?
- How does an employee add a new item to the menu?
- What happens if the restaurant sells out of something?

In this phase you need to be ANNOYINGLY detailed.

Spending the time here will help bring the big picture of your app idea into focus. When you take your idea and this user journey to your designer they will know exactly

how many pages need to be laid out and what needs to be on each page.

Your developers will be able to see what dots need to connect and where.

If we compare building your app to building a house, this part of the process is the BLUEPRINT.

Imagine if your foundation is already poured and complete and you've already framed in the first level... and then you approached the general contractor and said, "Oh I forgot that we need to have these 3 extra rooms on the top floor... can we push the foundation out 20 feet?"

They would look at you like you're crazy right?

This is what you want to avoid as you build your app.

The more detailed you can get when you are planning your features, the more efficient your build process will be. The last thing you want to do is ask your back-end developer to "push the foundation out 20 feet" because you forgot about a crucial feature.

Take your time on this and get it right.

How to test drive your app without writing a single line of code.

This is my ABSOLUTE favorite part of an app build. This is the part of the process where all of these abstract ideas, features, and flow charts start to feel real.

When we work with our clients, we never write a single line of code until we have a Clickable Prototype in our client's hands for a few days or weeks.

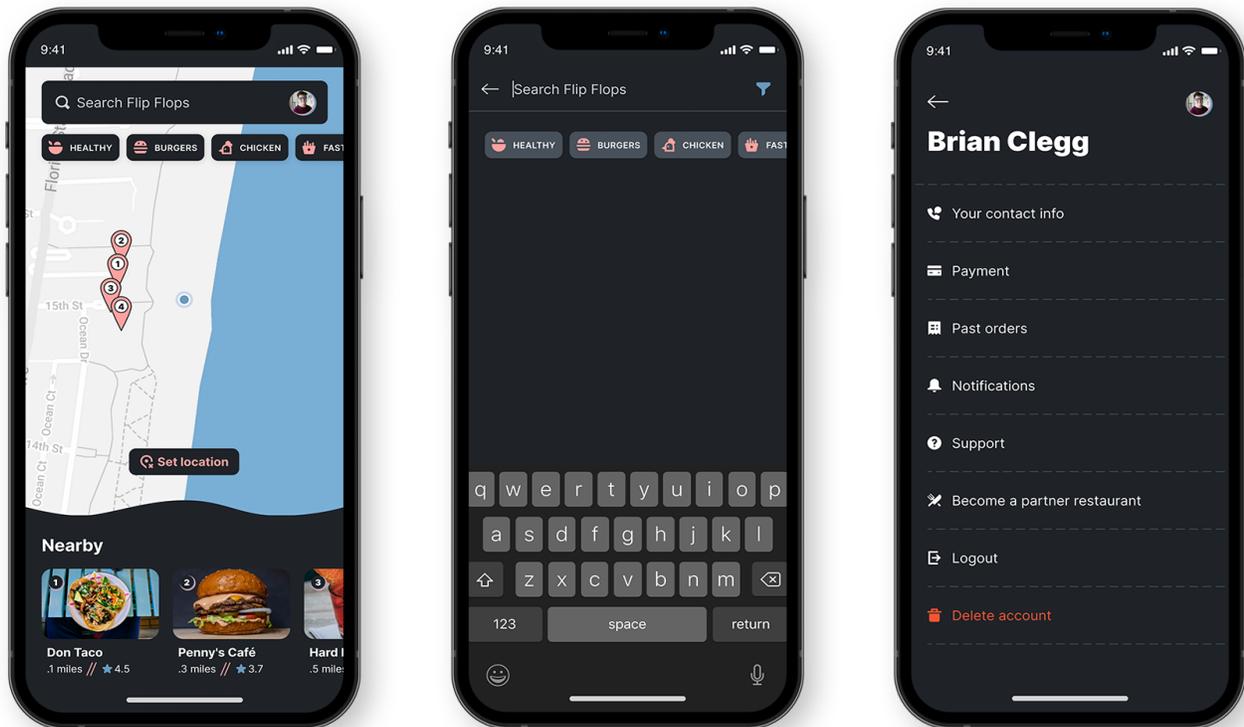
A clickable prototype is basically a completely designed version of your app, built in a program that allows you to simulate what it's like to use the app.

We use a design program called Figma that allows our designers to connect certain buttons, forms, and listings with other pages in the design file.

Then, they are able to adjust the dimensions of the design files to match the exact phone our client uses.

Let's say you have the iPhone 12 with the 5.42" screen. Your clickable prototype will be built for that exact device so you can see exactly how the REAL app would look and function on your phone.

These are some actual screenshots of the clickable prototype for that food delivery app I told you about earlier.



This part of the process is so cool because it smokes out all of the half-baked ideas. When you can actually see, feel, and interact with the designed version of your idea in the palm of your hand you'll notice things you didn't think about on the feature list or the user journey.

And here's the best part: You haven't paid to write any code yet! Making changes and solving problems at this stage is a fraction of the cost.

If you take one thing from this book it's this: Build your clickable prototype and test the crap out of it before you invest in 1,000 hours of code.

Have the prototype on your phone and show 20 people. Let them cruise around your app and see where they get stuck. Make revisions. Dial it in.

Nailing it and perfecting your idea at this stage of your build will save you so much time, effort, and money it's not even funny.

Writing code: Proceed at your own risk and peril.

Once you've spent some time with your clickable prototype, you have solid feedback from some future users, and you've refined your vision, you're ready to start writing code.

I want to give you a disclaimer: This part of the process is the most expensive. You really want to have your ducks in a row before you start writing code because you don't want to back track.

You'd be surprised by how fast you can blow through \$10k on code that could be completely scrapped if you change directions.

If you have 4 developers working on your project full time at \$120 an hour, it only takes 2.6 days to blow through \$10,000.

Am I trying to scare you? ABSOLUTELY.

Don't waste time at this point of your project.

Don't start coding if your idea is still fuzzy in your mind.

Don't start building features if you're still not sure if that integration you want is even possible.

And work with a professional!

If you aren't a technical person, that's ok, but you'll need to hire someone technical who can speak nerd and solve problems with your developers. Shameless plug: This is literally why I started Strides.

I noticed such a huge gap between the motivations of founders and developers — it's like they were talking completely different languages. Our goal at Strides is to help founders like you refine their idea and "project manage" the build so that really talented developers can do their thing and build your app in record time.

Writing code: Milestones and stand-ups.

Ok now that I've scared the crap out of you, let's talk about how your build will work when you have everything all dialed in.

Most qualified developers will work on your idea in sprints.

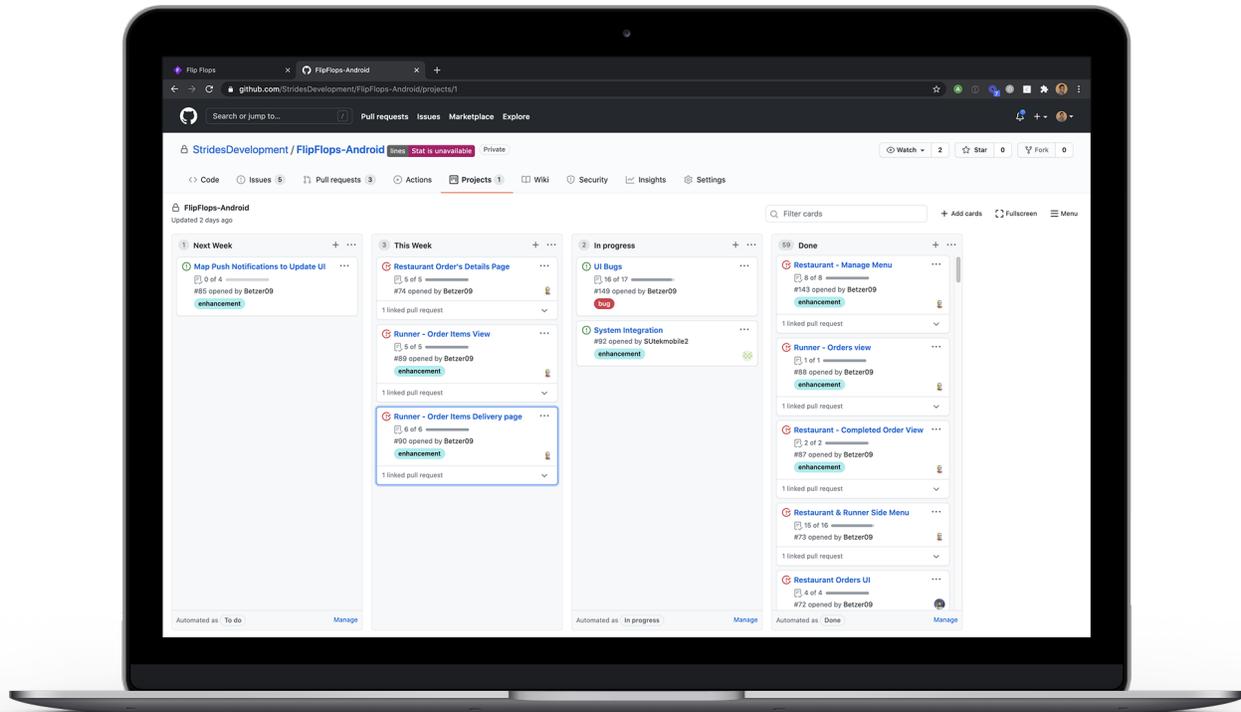
At Strides, once we start coding we make it a goal to complete a new feature every single week. This means that every week we have something new to show. Also, we typically start with the parts of the app that are the MOST unique and make the app stand out.

For example, if we're building a food delivery app, we're going to knock out the order screens first. The login, account management, and settings screens would be closer to the end of the project for us.

We want to get the exciting stuff built first because this creates momentum on the project.

The way we manage this part of the process is through a kanban board. A kanban board is an agile project management tool designed to help visualize work, limit work-in-progress, and maximize efficiency. It's really helpful for teams to establish order in their daily and weekly work.

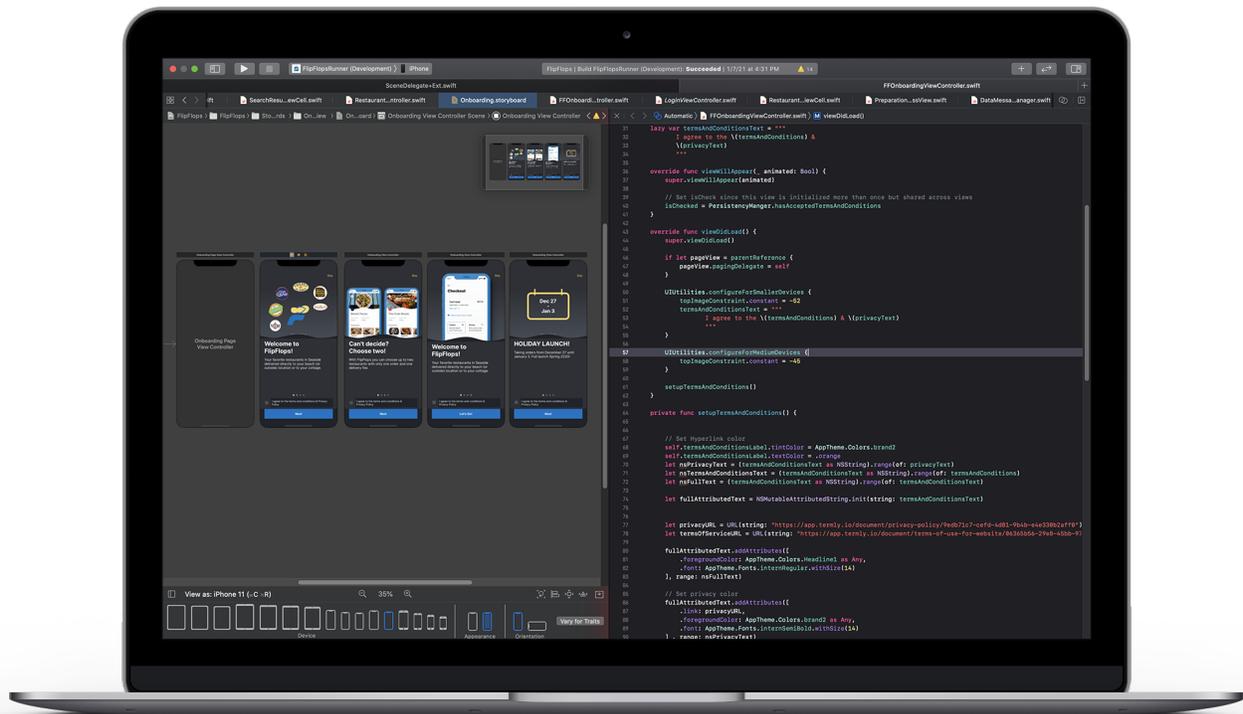
This is a kanban board for a project we recently worked on:



Once you have your project mapped out into clear chunks like this, it really comes down to writing code week over week until your app is done.

As far as the actual coding process, this isn't nearly as exciting as the clickable prototype part. The work you're paying for really just looks like a screen from the Matrix.

Here's an example of a project we worked on recently. On the left, you see the visual builder and on the right, you see the lines of code that make that picture come to life.



Testing and quality assurance.

As you're finishing features you gotta test them out to make sure everything is working.

This part of the process is simple but it's not easy. Basically you need to purposely try to break your new

app and this means that you need to test every feature under every possible circumstance.

- Does it work when you log out and log in with a different account?
- What happens if you're submitting your order and your data connection is spotty?
- Do the screens look right on a small device screen?
- What about a large device like an iPhone Pro Max or Android Galaxy Z Fold 2?

As you break your app and you find bugs, these bugs get logged and your development team will need to fix those issues.

“Going Live” and submitting to the app store.

Your strategy is locked in, your designs are done, your code is written, you've tested every feature and squashed all the bugs.

Now you're ready to launch your app. This is when you submit your app to the App Store for iOS and the Google Play store for Android apps. If you're building a desktop/web application, “launching” might just be giving your customers the URL.

Here's the thing, when you launch your app on the App Store or Play Store you have to be approved. There are lots of guidelines that you need to follow if you want the powers that be to allow you the opportunity to put your app in front of people.

There are also restrictions that each App Store has in place. If you violate any of their terms your app won't go live.

Here are a few of the things that will result in your app being rejected:

- Technical issues with crashing or bugs
- Poor performance
- Privacy issues
- Broken links
- Hardware and software compatibility
- Missing standard functionality
- Use of payment mechanisms other than "in app purchases"
- Your app is a duplicate or copy of another app
- Your app has placeholder content
- Your title or app description is misleading or inaccurate

- Poor UI or UX
- Incomplete information

If your app is rejected is that the end of the world?

In short, no. It is pretty typical when submitting an app to have it rejected 2-3 times before getting an approval.

Having an app rejected for having placeholder content, for example, isn't the end of the world. This is something you fix and then you submit your app again. This is an easy fix.

There are issues that can give you more problems. Let's say for example, you are submitting your app and you didn't know that the in app purchase system is the only authorized payment system in your category. You've built your entire app and your business plan around taking credit cards payments in the app. Guess what? Most likely you'll be rewriting a LOT of code and changing your business model if you want to have your app in the app store.

Getting your app rejected a few times as you launch is normal and it's all part of the process. HOWEVER, this is one of the biggest reasons it's so important to work with people who know what they are doing.

Working with a “cheaper” developer who doesn’t know Apple’s guidelines can end up being very expensive in the long run.

Part 2 Wrap Up:

This is a picture of an iceberg.



Your app is like this iceberg. 90% of the work is below the surface. Most of the effort will be spent on things that your user will never even see or notice in the finished app.

Successful apps don’t just happen — they are built by founders who follow a process that works.

Don’t shoot yourself in the foot — do what works.

Part 3: Launching your app.

Your plan for marketing your app might be just as important as your plan for building your app.

I love the startup culture that's so popular online right now... Personally, I love Gary Vee, and Grant Cardone, and all those guys that tell you to go out and chase your dreams.

HOWEVER... There is a popular line of thinking I see all the time on Instagram and LinkedIn that is incomplete and in my opinion DAMAGING to new app founders.

It's the idea that you can make your product/app so good that the product itself IS your marketing.

“Make something that people want to talk about and the only marketing you’ll need is word of mouth!”

I’m completely on board with making something so good that your users become passionate about it. But if that’s the ONLY thing you’ll be doing to drive app downloads, you’re going to be sitting there on launch day (and probably for the next 6 months) fighting for every single download.

You can only text your mom and your 3rd cousin so many times about your new project before they will block your number.

Have a basic plan for getting users. Know how you’re going to make money. Understand how you can afford to PAY to get new users.

Build a financial model and understand where your revenue will come from.

Something you’ll want to figure out early on in your build process is your revenue model.

I hear a lot of founders avoid this question based on more bad advice from social media. They watch The Social Network and hear Mark Zuckerberg talk about

how he doesn't want to let making money distract them from making a great product and they think, "Yeah me either!"

Don't be solely money focused — focus on solving a big problem and yes, you can figure out the money part later, but you should have an IDEA of how you'll make money before you invest in building your app.

Here are 5 basic monetization strategies and some very simple math for what it would take to generate \$1,000,000 in revenue with that strategy. Pick the one that matches your business model.

Paid app:

This is the most straightforward model. You make something that people would be willing to pay \$4.99 for in order to download and use.

If you sell 200,401 downloads at \$4.99 each, you'll make a million dollars.

Subscription:

Let's say you have something that would be valuable to a user month after month (think Netflix or Daily Burn). Maybe your app is worth \$7.99 per month and once users download it they will stick around for 9-18 months.

If you have 10,430 subscribers paying you \$7.99 per month, you'll make a million dollars a year.

Serving a bigger purpose within your company:

Maybe you are a high-level business consultant and you're looking for new clients for your management consulting practice.

You come up with a really useful idea that your target market would love. You could give away your app for free to get leads if you have a clear way to convert those leads into paying customers down the road.

- Let's say 5 out of 100 people who use the app would be interested in having a call with you about your services.
- Let's say you close 50% of the potential clients you speak with on the phone.
- Let's say your consulting packages are \$5,000 per month or \$60,000 per year.

If you get 680 people to use your app, you would create 34 new phone calls, close 17 new clients and make a million dollars in revenue over the next year.

Transactional:

Let's say you are creating an ecosystem where users can pay for services through your app in exchange for paying you a percentage of their transaction (think Uber, Lyft, or a classifieds listing app).

Let's say you make 18% of the revenue generated from a sale on your app.

If you process \$5,555,556 in transactions and you take 18%, you'll make a million dollars.

Business Process:

Maybe you are building an app for your internal team and it's not for the end consumer. If your app is built to streamline a business process with your employees, add up how many hours per employee this could save for your company.

If your app saves one employee 5 hours per week and the average employee at your company is paid \$35 an hour, then your app saves your company \$9,100 per year, per employee.

If 110 people in your company use the app it will save your company a million dollars per year.

A quick note...

None of these simple estimates include the fees you would have to pay to Apple if your app makes money through in app purchases. This is meant just to get the ideas flowing in your mind and have you start thinking through things on a bigger picture.

Have a plan for getting new users.

Now that you have an idea of what it would take for your app to generate some significant revenue, now it's time to start thinking about how many new users you'll need to get in your first year to hit your goals.

Look, there is a LOT we could cover in this section... But for now let's just keep it simple:

Think about who your target audience is and where they hang out? What sites do they visit? What books do they read? What conferences do they attend? What's a typical day in their life look like?

Once you know where to find them, get your app or idea in front of them!

This doesn't mean you JUST blast them with "download my new app" messages... Also talk about why they should care about your app.

Identify the pain points that resonate with your audience and communicate how you are solving them. Explain in layman's terms how you are going to make their lives easier. Be precise — you don't need everyone to click on your ads or talk about it — just the right people.

Test out what works. Try a few pain points and see what brings the people in. Pay attention to data and change up your strategy accordingly. Test out your messaging and positioning until you get it right- keep going!

Don't miss the easiest place to start...

Don't forget to talk to any and all of your first customers and ask them why they signed up with you. Use their feedback to make messaging decisions. Mostly likely you will be able to find consistencies and apply that customer feedback to marketing and app development/iteration. Talking to customers is the best thing you can do.

Let's see this in action:

One of our clients — EquineTrader — has a platform for horse lovers to buy and sell horses.

How do they get users? The founder knows all of the big breeders across the country. He worked with those breeders to list their horses on his app and guess what? When people call up that breeder and want to see a photo of the horse, the breeder tells them to look at the app. Partnering with big influencers in your space is a fast path to users.

If you are selling a meditation app, maybe you want to run Facebook ads towards people who like to meditate and are passionate about mindfulness.

If you're selling a food delivery app, start with the restaurants and run Facebook ads within a 5 mile radius to attract new customers.

Part 3 Wrap Up:

Don't be that founder who launches their app and then starts thinking, "Oh, how should I market this?"

A common misconception we frequently hear is that you can't start marketing without a finished product. Plenty of smart founders start generating awareness and sales on things that aren't complete!

The more strategic you can be and the more you can align your app with a specific target audience and solve

a specific problem for them, the easier it will be to know how to reach them.

The advantage to solving specific problems for a certain type of person is that there will ALWAYS be early adopters out there who are more than willing to try your beta product, give feedback, talk to you, and be the champions of your product.

The best thing to do is to just start moving — yeah, make sure you're not making empty promises but don't wait until you have all the puzzle pieces perfectly in place either.

Section 4: Building your team.

So how many people do you need on your team?

Man, this is a tough question. The answer is that it really depends on your project. Think of it like this:

*“How many people does it take to
make a pizza?”*

For sure you could find ONE person who could manage all the steps and make a decent pizza right?

Let me ask you another question:

“How many people does it take to make a successful pizza restaurant that serves hundreds of people a day?”

Ah... different question altogether right?

Making an app is similar to making a pizza — you could for sure find a one-man band who could make design, build, code, test, and launch your idea for you.

But making a SUCCESSFUL app? That’s a different story.

The apps that are launched by 1 or 2 people and go on to have major success are incredibly rare. And in those rare cases, the founder of the company is almost always a designer or developer.

Now that we have a clear picture of what the steps look like from idea to launch, let’s detail out who does what, and who you’ll need on your team.

Project Management.

Project Management is all of the administrative stuff that it takes to keep an app development project on track.

It can be basic stuff like setting up meetings and managing lists of features and potential features or it can be more complicated like reading through technical documents from your developer and getting all of the players on your team to speak the same language and have access to the same vision.

Who does this for you?

- YOU. You might be doing a lot of this at first
- A project manager
- A technical producer

Strategy.

This is where you figure out why your app even exists — and you definitely need a better answer than, “I think it’s a good idea!”

It’s worth it to pay for some help to get clear about what problem your app solves, who it’s for, and what makes it different from everything else out there.

Who does this for you?

- A marketing agency
- A freelance marketing specialist

- A sales copywriter
- YOU. You might need to do a lot of this work yourself in conjunction with mentors and a coach. The ultimate vision for your idea should rest with you — don't try to outsource this.

UX & UI design

This where a designer (or several designers) mocks up every single screen in your app.

A lot of people get confused about the difference between UX and UI.

UX stands for “user experience” and focuses on the overall usability of your app. Is it intuitive for your users? Are things where they are supposed to be?

UI stands for “user interface” and focuses on the final design of the actual interface that your user sees when they use your app.

Your UX designer worries about how your product makes a user feel, and how easily they navigate your product. Your UI designer worries about how cool the hover state looks on a button or the ease-in-out of a drop-down menu. They also ensure your application is accessible. A lot of times these roles overlap with one another, but

as you can imagine this is a critical component of the development of your app.

Who does this for you?

- UX designer (specialist)
- UI designer (specialist)
- A single designer who does both (generalist) although, I'd recommend avoiding generalists and work more with specialists.

Branding

This is a fun stage in the process and it's where you develop your logo, colors, visual identity, what typefaces you use in your design, and your overall style.

Who does this for you?

- Branding agency
- Freelance designer
- Multiple designers

Overall Build

A typical mobile app project is made up of three integral parts: back-end/server technology, API(s), and the mobile

app front-end. In the build stage, developers create code, integrate analytics, and introduce features.

Who does this for you?

- Front-end developer
- iOS developer
- Android developer

Testing

This is where you get feedback from your initial users by studying analytics and analyzing in-app behavior and consumption patterns to enhance your product.

Who does this for you?

- Your project manager
- Your developers
- YOU
- Your friends
- Your beta customers
- Your mom

Marketing

Finally, it's time to promote your app to users. This is where marketing campaigns and techniques are employed to communicate with users pre-launch, launch, and post-launch.

Who does this for you?

- YOU
- A hired marketing manager
- A marketing agency

Part 4 Wrap Up:

If there is one thing I could leave you with as you start to put your team together it's this: Building your app isn't a one-person job.

Work with specialists and find talented people. Talented people WANT to work for visionary founders.

You are the conductor. Start putting your orchestra together!

Getting started.

If you're reading this now, I haven't scared you away. :)

It was tough to pack years of experience in this industry into a single book but I hope this gave you a high-level understanding of what it takes to build and launch a successful app.

As you can see, it's not easy. There are a lot of steps to take and there are a lot of talented people you'll need on your team if you want to do it right.

But here's the thing: If it was easy, then everyone would have 10 apps and chances are, someone would already have your app idea in the market.

The fact that it's hard can be your advantage.

If you're reading this now and you're not scared away from the entire process then you might be someone we want to work with at Strides.

If this is you, I want to tell you about something really cool we offer that will help you get the ball rolling on your project.

Here's the thing: So many people think that building their app means "writing code."

Writing code is super important — it's one of the big things we do here, but I'm hoping that you're getting the idea that writing code is only PART of the process.

I noticed early on with our clients that they needed help getting their idea finalized. They needed some expert guidance in thinking through their target customer, prioritizing which features would be the most important and which should be built first, and they needed help getting their prototype wrapped up so they could start testing their app and showing it to potential users.

So I'm pretty excited to roll out a new program that helps you do everything you need to do PRE-CODE.

We call it the App Builder Accelerator.

This is a program designed specifically for first-time app builders and we take them through the process we've outlined in this book. We work with you over 4-12 weeks through a series of workshops and you'll walk away a COMPLETELY validated concept and a clickable prototype.

We also build-out all of the business strategies that you'll need if you are trying to get your company to commit a budget to build the app or if you are getting an investor to help you build it.

This process does cost \$5-\$20k total but it will be the best money you ever spend. This process saved one of our clients over \$50k in development costs alone. Another client we worked with came to us when they already had code in place that we had to end of scrapping. We estimated that this process would have saved them something like \$200k.

Here's a few things we'll help you with in this process:

- Researching your idea and finding competitors and other apps doing similar things.
- Helping you refine your feature list.

- Meeting with developers to prioritize your features and make sure you're committing to building things that are feasible on your larger budget.
- Working with our accounting teams and building your business model, customer acquisition model, and your "unit economics."
- Building your user journeys and thinking through all the screens and features in your app.
- Building a clickable prototype that you can play with and show to potential users or investors.
- Creating go-to-market plan with a strategy for getting leads, where you can advertise, and how to get in front of new customers.

It's a lot... You can come to us with your idea on the back of a napkin and then walk away with a full business plan.

After that, you can take your project to your team to develop it, or you can work with our team to develop and launch your project.

This really is the best first step you can take if you're serious about turning your idea into a real thing.

Is this you? Ready to get started?

If this is you, let's talk!

We offer a free strategy call to anyone who isn't scared away by the process. :)

Reach out at this link below and get on our calendar.

We'd love to chat with you about your idea, see if we'd be a good fit together, and give you some ideas on your app!

Get your free call at strides.dev/call

Thank you!

It's been fun hanging out with you.

Thanks for reading this and for being here. Feel free to pass this along to anyone who would find it useful and make sure you're following us for more updates.



Follow me on LinkedIn for more useful tips & ideas
<https://www.linkedin.com/in/betzeraustin/>



Follow Strides on Instagram
<https://www.instagram.com/strides.dev/>



Ready to book a Strategy Call? Head over to our website!
Get your free call at [strides.dev/call](https://www.strides.dev/call)