

SideProject Book

Table of Contents

<i>Checkout Helper, Mark Rickett</i>	4
<i>TweetingMachine, Tom Buck</i>	9
<i>Trism, Steve Demeter</i>	15
<i>Domainr, Eric Case</i>	18
<i>Enigmo, Brian Greenstone</i>	21
<i>BatteryBar, Chris Thompson</i>	23
<i>Fantasy Monster App, Jerry Shen</i>	28
<i>Bingo Card Creator, Patrick McKenzie</i>	30
<i>HoursTracker, Carlos Ribas</i>	35
<i>MockingBird, Saikat Chakrabarti</i>	37
<i>Simple Physics, Dummy Defense, Andrew Garrison</i>	40
<i>Feedback Army, Raphael Mudge</i>	44
<i>Gratitude Journal, Carla White</i>	50
<i>Wedding Lens, Justin Chen</i>	52
<i>Droid Secret Tips, Tim Mackenzie</i>	55
<i>Textual App, Jeff Weisbein</i>	59
<i>DNSimple, Anthony Eden</i>	61
<i>Ventrilode, Charles De Meulenaer</i>	64
<i>Are My Sites Up?, Richard Felix</i>	68
<i>Photo 365, Benjamin Hsu</i>	70
<i>RideWithGPS, Cullen King</i>	76
<i>Dream Track Nation, Tiny Plane, Mathieu Roy</i>	80

<i>Project Bubble</i> , Stu Green.....	82
<i>The Little Crane That Could</i> , Bram Stolk.....	85
<i>Artsy Editor</i> , Stephen Ou.....	89
<i>Debt Snowball+</i> , Derek Clark.....	91
<i>Bidsketch</i> , Ruben Gamez.....	93
<i>ParkWhiz</i> , Jon Thornton.....	97
<i>Monster Coloring Book</i> , Christopher Taylor.....	100
<i>Visual Website Optimizer</i> , Paras Chopra.....	103
<i>Living Earth HD</i> , Moshen Chan.....	106
<i>Gumroad</i> , Sahil Lavingia.....	111
<i>DragonDrop</i> , Mark Christian.....	113
<i>Unlock With Wifi</i> , Ben Hirashima.....	117
<i>Sticky Notes</i> , Mark Peterson.....	121
<i>HackerNode</i> , Sumeru Chatterjee.....	123
<i>DotNetInvoice</i> , Rob Walling.....	125

Checkout Helper



iPhone

Idea: Automate the process of adding up multiple items, taxes, and shipping, to save time in calculating totals for orders.

Project Maker(s): Mark Rickett

Download App: <http://itunes.apple.com/us/app/checkout-helper/id421049699>



Hi Mark, thanks for sitting down to talk with us. Tell us About your background prior to starting Checkout Helper.

Mark: My degree is in Management Information Systems. I've wanted to be a software developer since High School so in college I focused on information systems thinking it would give me a background in business also. Unfortunately, the track I took in college did not really focus on learning languages or a lot of practical application development, so I taught myself PHP in my free time. I got a PHP programming job straight out of college as a Jr. Developer managing a million dollar online ordering system.

My basic fundamentals in programming helped a lot while developing for iOS. Objective-C was unlike anything I had ever seen before, but a strong background in Object-Oriented Programming and MVC frameworks made it fairly easy to pick up and get the ball rolling.

Already knowing how to code must have given you a leg up when the app store started becoming popular, had you worked on any side projects before Checkout Helper?

Mark: I've always tinkered with my own side projects, and luckily, my employer is cool with it—they encourage us to learn new things and grow our skill sets on our own time. Most of my side projects were jQuery plugins, Wordpress themes and plugins, and small PHP applications that I used for various purposes. My most popular side project by far is the Nike+ Wordpress plugin (<https://wordpress.org/extend/plugins/nike-ipod/>) that lets you display your runs and stats for the Nike+ running system in your Wordpress blog's sidebar.

I had watched the App store mature over time and decided I was going to learn how to program for the iPhone. My goal was to make back the \$99+tax developer fee every year, so I went about brainstorming ideas for a simple app.

What did you come up with?

Mark: My wife sells jewelry at people's homes and I decided to make a point of sale application that would help her automatically add up customer orders and add tax and shipping. It was a simple idea and I thought it would save her taps on her iPhone's calculator app if these things were done for her automatically and all she had to do was enter the prices.

I found out that there were not very many apps that catered to Direct Sales. One here or there, but they all had horrible user interfaces and didn't do exactly what I wanted. So I made one myself.

Differentiators I'd say were not a big deal simply because there were no apps that catered to the market I was targeting.

And then you decided to make it, how did you get started?

Mark: I built a REALLY rough prototype and released it in the App store for a reduced price. This helped me figure out the app release procedure and also started making me money. Once I realized that yes, there really was a market for this kind of app, I went about revising it and making it better over time. Iteration is the biggest thing you can do. Keep making it better.

How did you find the time to put in the work with a full-time job?

Mark: This is and will continue to be a part-time hobby for me. My wife worked primarily in the evenings and we don't have kids, so it was easy to find time to myself in the evenings to work on my side projects.

Did you have any initial costs?

Mark: I already had the Mac, all I needed to do was pay the developer fee to start testing on my device. I started developing my apps in the simulator (free) and once I had a working prototype, I paid the development fee to actually test and publish the app.

How would you recommend someone who has a full-time job that wants to build a side project get started?

Mark: Carve out the time to work on it, no matter what. Stop playing in the softball league at work... stop going to the movies... in fact: stop watching TV altogether. That will free up a ton of time for most people.

How did you decide upon your final business model?

Mark: Revenue model is Paid+In-App purchases. Since I'm targeting business users, they view it as a business expense and are therefore willing to pay more for the app. I then monetize existing users by offering in-app purchases to customize the way their app looks.

Checkout Helper is consistently in the top 60-100 paid and top grossing iPhone apps for the business category in the US (it's pretty volatile). I jump on and off the leaderboards for Canada, but I don't get many sales in other countries.

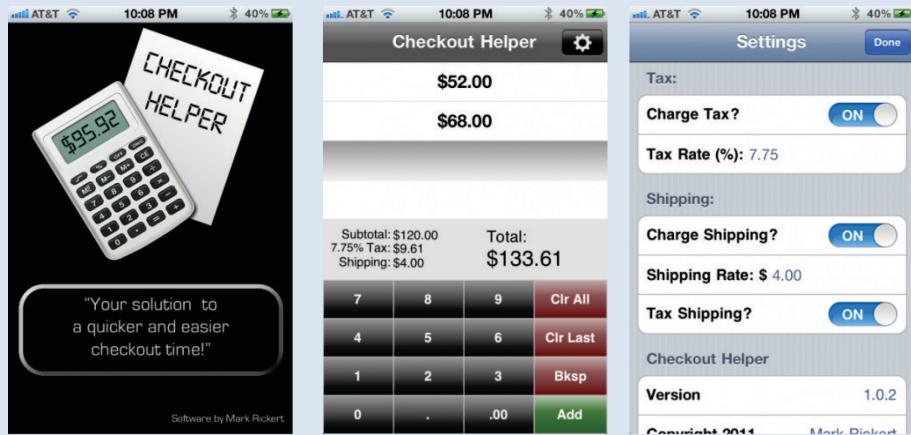
Since launch, the app has averaged about \$47/day but more recently, in the past six months, it's averaged about \$65/day (net profit – after Apple's cut). About a 15% of my users make an in-app purchase, which is stellar, and putting that mechanism into the app has made almost \$1,000 in the last 8 months. On my single best day, Checkout Helper brought in about \$160.

When you launched, how long was it until you saw your first sale, and where did it come from?

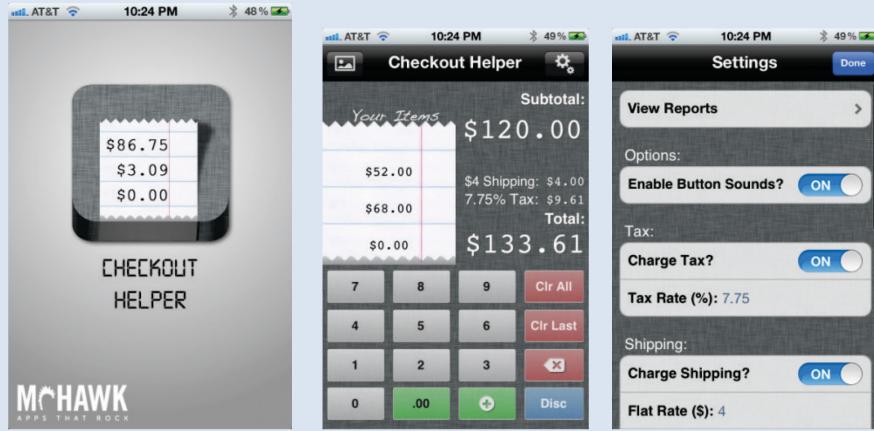
Mark: I rely completely on word of mouth and organic growth. I told a few friends and had 1 purchase on the first day. For the first 3 months or so I had a steady 1-10 purchases a day. Immediately upon releasing version 2.0 I started seeing growth because I started a Facebook page and had people start talking about it online. Facebook was a natural marketing channel since my target demographic is women. Checkout helper is averaging about 25 purchases a day now.

How long did it take to build your initial version, MVP?

Mark: The MVP was built and released in a few weeks.



Version 1



Version 2

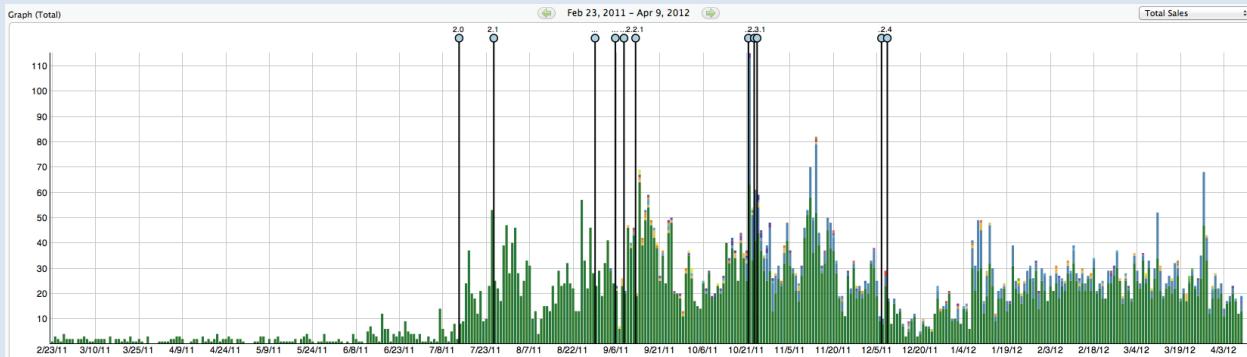
What parts of development did you do yourself?

Mark: I did all development and graphics on my own. No contractors or outsourcing. I wanted this to be completely my “baby”. In more recent apps, I’ve contracted out the design of the icon and loading screens, but still do all the in-app artwork and user interface design myself.

How was growth in the early days?

Mark: Growth started out really slow. Probably because the MVP was not very good, but once I iterated enough time to have a 2.0 release, the sales went up from there.

Here's a graphic of sales:



What has been your biggest surprise so far?

Mark: A few,

1. I can't believe I found such an under-served market.
2. People will pay for in-app purchases, even though the app isn't free.
3. I'm actually making a decent income off this.

What has been your biggest mistake?

Mark: Not doing it sooner? I watched the App store evolve for a few years before I tried my hand at it. Also, one time I released a version without testing it when there was no network connection. Turns out something I added crashed the app when it couldn't reach the internet. Whoops. I had to scramble to release an update. Things like that can get you 1-star reviews.

What tracking tools do you use?

Mark: I use Flurry for tracking and analytics. It's a fantastic service and provides some great metrics such as user retention, session length, sessions over time, etc. It also will tell me about app crashes when they happen.

How is your project doing now, and what are your plans for the future?

Mark: It's mostly in cruise control – generating passive income for me. I have plans for a few new features, but it's about 95% done with the app as it sits right now. Once I add in the last few features I don't anticipate any more updates to the app other than iOS compatibility upgrades.

What advice would you give to a developer that is wanting to work on their own side project?

Mark:

- Build it and release as fast as you can and then iterate. It doesn't have to be perfect the first time.
- Target a niche, underserved market (don't try and make the next "Angry Birds").
- Create something functional that solves a real-world problem, unless you're making a game... and in that case: just make it so that kids will think its fun to play.
- Don't stop with one app – create multiple apps for an even higher revenue stream. Checkout Helper is only about 2/3 of what I make in the iTunes store.
- Don't underestimate the patent trolls. Incorporate in your state as an LLC. The process to switch your individual developer account over to a company with Apple is a hassle, so do it from the start.
- Decide if you want your apps to be universal and if so, make them universal from the start. It's a pain to add in iPad support after the fact.
- Don't rely on ad revenue if your app is primarily a utility app. Ads work for high user-engagement time apps like a Facebook client or a game, but unless you can keep a user in your app, make it a paid app and not an advertisement-based revenue model. I have one app that had ads in it and it's made a whopping \$12 so far (about 50¢ an hour).
- Don't get discouraged by negative reviews... unless there's a legitimate problem.
- Don't underestimate the power of metadata in iTunes. Keywords are not seen by the public but

- have a huge impact on user searches. Come up with excellent and relevant keywords.
- Have fun building your apps!

TweetingMachine



Browser

Idea: Create an app to let users schedule their tweets.

Project Maker(s): Thomas Buck

Website: <http://www.tweetingmachine.com/>

A screenshot of the TweetingMachine website. The header reads "Welcome to TweetingMachine". It features a "Try For Free!" button and several promotional banners. One banner highlights "The Ultimate Twitter & Facebook Tool - TweetingMachine - only \$19.95/month". Another banner says "Get More Customers With Less Work! Make your Twitter campaigns quickly and easily. Try FREE now and get a 30-day money back guarantee." The footer includes social media links and a "Sign in with Twitter" button.

Hi Thomas! Thanks for talking to us. First off, how did you get into technology?

Like many other techies, I've had an interest in technology from an early age. When I was five or six, my brother had a 64 that he'd let me mess around with. He'd teach me how to load games, run the odd BASIC command to change the screen colours and so on. Really basic stuff, but I found it incredible that I could manipulate the magical box plugged into the TV.

A few years later, I was given an Atari ST for Christmas. This is what got me really started; it would be the only computer in my household for the next eight or so years. As I grew older, I moved from just playing games to exploring how they worked, whether by editing resource files and replacing the graphics with my own, or trying out the absolute basics of programming in BASIC.

I'll be honest; I wasn't one of those super whizz-kids who understood programming from an early age and wrote assembly routines and games in their free time. I was much more of the "I wonder what happens if I make this change" type of kid. That said, this early curiosity has left me with a couple of good talents in later life: I can very quickly figure out codebases and systems that are entirely new to me; and I'm pretty hot at tracking down obscure bugs.

I've also had more than my fair share of luck. My school got its first internet connection back in 1995, right when I was turning 13. I taught myself HTML, created a few websites about the Atari ST and had fun breaking my head trying to get CGI programs working in Perl.

I got my break when I turned 16 in 1998. One of my sister's friends gave me a summer job working at a web development firm. The boss there, Chris Love, gave me an opportunity to prove myself, and I grabbed it with both hands. I would continue to work on a part-time basis for his company until I turned 18, moved to London and started working there full-time.

These were still the early days of development using Linux, Apache, MySQL and PHP (LAMP). It would be a couple of years until the Smarty templating system would be released, and so we were dealing with a lot of custom-written code, or our own internal development tools. I had to code things like user sessions on their own, which gave me a real insight into how a lot of core technologies like cookies worked – and meant that I picked up some fun debugging skills.

Over the next decade, I would work for an SMS aggregator and applications provider, and eventually head up the development team for a company that monitored online auctions for various big brand names. This meant lots of fun with image recognition and managing staggering amounts of data.

I've been fortunate enough to always love my work, and have undoubtedly had luck on my side. But without having had made the initial effort, there would've been no luck to exploit. This is something that I

think all would-be entrepreneurs need to drum into their heads: you can't get lucky without having something to get lucky *with*.

What were you doing before you decided to build a side project, and how did it come about?

I moved from my native London, UK, to Warsaw, Poland, at the end of 2009. This was for purely selfish reasons; my girlfriend (now wife!) and I were decided who was going to move where, and I wanted to be the first to experience living in a foreign country.

The first couple of months were fine, easy living; I'd brought over my savings, and never really having had a proper, several-week-long vacation in well over a decade, I was now taking the opportunity to relax.

My vague plan was to either find some freelance work from contacts in the UK, or work in the local jobs market. Sadly, between the financial crises at the time and my own freelancing inexperience, I wasn't having a lot of luck finding work.

I started looking for work on RentACoder.com (now called VWorker.com). Sadly, for reasons I still don't quite understand – that, or more likely, I'm refusing to recognize my own stupidity – I thought that because I was based in Poland, I should charge low rates, not recognizing the true value of my skills. So, I was doing OK – not much money, and we'd have a few tight months over the upcoming year.

We were planning on going on holiday to India at the start of 2010, and a few weeks beforehand I had a thought: What if I could create one of those online web tools? It would only take a few new subscribers each month, and before I knew it, I'd be rich, rich, rich. I have the technical skills – the design won't be great, but will suffice – and from there life would be good.

I decided that in the three weeks before we left, I'd create something – anything! – get it online and accepting payments, and look forward to seeing a healthy PayPal balance upon my return to Poland.

My SaaS desires had been born.

How did you come up with an idea?

I was searching for work with my world-domination ideas floating my head, when I saw an advert on RentACoder. It was the typical nightmare request; a vague, two-line description, a budget request of \$50, and plenty of opportunity for both sides to get thoroughly confused. It was a request for a website that would let people schedule tweets on Twitter.

I thought, you know what, that sounds like fun, and pretty easy to do. And I wonder, if someone has this idea, and is willing to pay to have it developed, maybe there's money in that space?

A quick couple of Google searches later and it became obvious that people were making money with Twitter tools. I figured that there's always room for one more tool, and sat down to write it.

My main idea was that the existing tools were pretty clunky and unintuitive to use. I had to consult the help sections of competitors' tools a few times just to work out how to schedule a tweet! That sounded like absolute madness to me – in an ideal world, help sections should be available but not necessary.

TweetingMachine would be different: quick, easy, intuitive. A tool for "normal" people.

How did you get started?

When it came to getting started, I thought that coding TweetingMachine would be the hard bit. People use search engines, it can't be *that* hard to get decent levels of traffic, could it? Little did I know.

I sat down and worked out how this would all work. The technologies used would be PHP, MySQL, hosted on Linux with Apache. I'd come to really like the Kohana PHP framework (<http://kohanaframework.org/>) for its simplicity and how quickly one could knock up applications using it.

Next, I wrote down the rough database schema I would use, along with how I thought various parts of the system should work. For instance, I'd need a nice library for communicating with Twitter's API, and various objects for parsing tweets and so on.

I treated TweetingMachine like I would any other project; it's counter-productive to dive in with the coding straight away, you need to plan how things are going to work before you've wasted hours implementing a solution which won't do what you want it to.

That's not to say that I did everything as ideally as possible; I had an immoveable deadline to get this done before I headed off on holiday to India. There were a few late nights, and consequently less-than-optimal code quality. A few shortcuts here and there, to be fixed at a later date.

Did you work on TweetingMachine while you were doing your regular work?

For TweetingMachine, I made the decision to take a few weeks off from freelancing and work on it instead.

Aside from the opportunity costs of me not working on something else, I went with the cheapest VPS hosting I could find. This was Prgmr: <http://prgmr.com>. I can't recommend Luke, Megan, Nick and the rest of the Prgmr crowd enough – absolutely superb hosting, and I've had worse service from much more expensive companies.

Since TweetingMachine, I've created a few other web applications, the most recent of which is Interactwive (<http://interactwive.com>). At the time I was working full-time for a firm of consultants. My free time was quite limited, and my wife wouldn't have been that happy had I ignored her entirely either. So I went for the best option I could: work a few evenings a week, maybe 20 hours total over 7 days. If I was travelling by plane, or on a long train journey, I'd take my laptop with me and take advantage of the available free time.

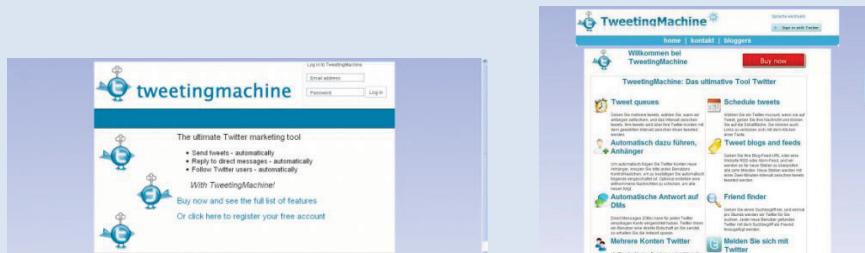
I was very tempted to hire a freelancer or two to get the work done for me. It quickly became apparent, though, that I couldn't afford the same quality of work that I would like – or write myself - and so I settled on sacrificing my free time instead.

What is your business model?

Initially, I planned on having three different levels of monthly subscription: one really minor that only allowed a single Twitter account; one in the middle that had a few Twitter accounts; and a premium option. After selling a few "middle" option subscriptions, I decided that the different subscription levels were too confusing, so scrapped the lot and came up with a single plan. I experimented quite a lot with the price, going from \$9.99/month to \$19.99/month, and even trying an annual price of \$19.99!

When you launched, how long was it until you saw your first sale, and where did it come from?

For the first six to nine months, TweetingMachine was a complete and utter failure. Whilst the tool worked well, and customers loved my responsiveness, conversion rates were minimal – and people using the free trial even fewer.

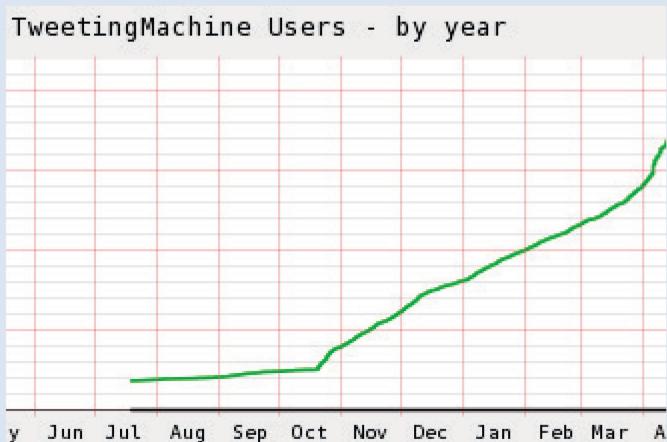


Early Version, TweetingMachine

The turning point came when a friend of mine suggested fixing my own useless attempts at making the

tool pretty with a couple of themes from ThemeForest (<http://themeforest.com>). I sat down for an evening, and a few hours later had a tool that, especially compared to its earlier design, looked cool and professional.

Here's the graph that shows the spike in free trial take-ups as soon as the new design was launched:



A TweetingMachine subscription at this stage cost \$9.99/month. I later experimented with an annual price of \$19.99/year – and had a fantastic December, simply incredible by my standards, somewhere around \$500 total revenues.

After a challenging couple of months at the start of 2011, I changed the subscription period from paying once a year to paying once a month instead. I haven't looked back since.

When it came to promoting TweetingMachine, I had a few ideas in my head: first and foremost, perform some basic SEO so that people searching for "schedule tweets" and so on will eventually see my site in their Google results.

Next, I started searching blogs for mentions of my competitors. I then contact the various blog owners, introduced myself, and asked if they'd be interested in taking a look at TweetingMachine. Once I launched TweetingMachine's new design, my success rate here went way up as well.

I experimented with offering an affiliate programme. I had very basic terms: for every sale you give me, I'll give you 50%, and will continue to do so for every month after that. Sadly, very few affiliates appeared to be interested in this offer, and it led to a fair few people spamming away on Twitter instead. I think that affiliate programmes can work very well, but you definitely need to keep an eye out for people employing shady practices, and be aware of how these might reflect on your reputation.

How long did it take to build your initial version, MVP?

The initial version of TweetingMachine was developed in three weeks. This had the vast majority of features that TweetingMachine still has to this day. Since then, in total I've performed around a month of development, so maybe 2 months in total.

And you decided to develop it all by yourself?

I handled all aspects of development myself. This included setting up the server, figuring out what was happening when I was managing to repeatedly crash Apache, learning a lot about optimizing code for small VPS instances and so on.

Overall though, my total cash outgoings were very small. \$9.99 for the tweetingmachine.com domain. \$20/month for hosting. \$40 or \$45 on themes (much later on). And the rest was sheer hard work!

How has design affected your app's performance?

As time progresses, consumers are expecting apps to look better and better. Themes can suffice for your initial version – if you, like me – have zero design skills, you **need** to be using them. After that, definitely get a designer on board. For my various apps – Interactive included – I have the conclusive numbers that normal people prefer better-looking apps.

I think there could be a real market out there for creating web versions of existing ugly apps, sticking a theme on them to make them look pretty, and gaining customers who are sick of working with apps that look like they were last updated in the Windows 3.1 days.

What has been your biggest surprise so far?

Quite how important it is for your application to look pretty. I don't think that an application has to go overboard with its design, but a bit of prettiness really helps. One application I really admire is Buffer (<http://bufferapp.com/>). Its design is simple, but elegant, it works really well, and their marketing is beautiful. They're the guys whose way of going about things I'm going to emulate this year!

What has been your biggest mistake?

Without a doubt, believing that "if you build it, they will come." This might happen once a lifetime, but so might winning the lottery.

No, when you have your idea, you need to go out and find people who might be interested in buying it. Talk to them, find out what they want and need, and then try to sell it to them. You can do all of this before you've written a single line of code.

What tracking tools do you use?

What I really, really should be doing now is a/b testing and experimenting with my traffic to see what happens. For the time being, I use Piwik for my web analytics (<http://piwik.org/>). It's a really nice and flexible system, and it's pretty easy to get it talking with the internals of your web apps, should you want to follow paths of individual members and so on.

I've played around with Ad Words for Interactive, and have used its conversion tools to gain some insights. Thoroughly recommended, and at some point I'll find the time to integrate these into TweetingMachine.

How is your project doing now, and what are your plans for the future?

TweetingMachine basically pays the rent now, although its growth has slowed – stagnated even – over the past few months. I have an idea of reducing its options and complexity and spinning off a consumer version for people with only one or two Twitter accounts.

My main focus for now is Interactive. Its growth has mirrored TweetingMachine's in a way. I launched an ugly version, got it working, made a few sales, launched a redesign, made more sales, and I'm now working out how I'll continue to market it in my free time.

A quick word on marketing: if you don't have precise profiles for the kinds of people buying your apps, you're going to be in trouble. Believe me on this – it's a recent revelation, and since addressing it, I'm finding it easier and easier to contact people who are interested in my apps.

What advice would you give to a developer wanting to work on their own side project?

Whenever I blog about working on a new idea, there are usually a good few comments from people waiting for the "right idea" to come their way. I think it's too easy to get focused on the idea, and not on simply creating something – anything!

If you want to create a web app, go and find an existing app that you really, really like. Work out who you'd like to be buying it, and how you're going to reach out to them.

Then set a date for launching, and find the time to code it. Stick to your launch date and get it out there. You'll learn more about web apps by launching than you ever will do from reading about other people's attempts.

And have fun! It can be a pretty lonely and frustrating experience at times. If you need to take a break away from it all, then do, and don't feel guilty that you're using your free time for leisure. Then come back with a fresh head, work on improvements, and see what happens ☺

Trism



iPhone

Idea: Build an indie hit game without the help of establishment.

Project Maker(s): Steven Demeter

Download App: <http://itunes.apple.com/us/app/trism/id284653044>

Tell us a little about how you got started in making games.

I have been writing code since high school, making games for TI Calculators. My games background goes back to the Atari 2600. My parents would always ground me from it when I would get in trouble, but once they saw the vast moneymaking potential of the internet, their story changed to "Ok, stay on the computer as long as you want, just please don't get TOO cranky".



What were you doing before you built Trism?

I was leading a team of developers for a large bank, writing their ATM software. At least, that's what I always tell the press. I really got my chops from ROM Hacking and Translating. After I graduated high school in '96, I got really into Japanese, because game emulators were just starting to come out on the internet and gamers quickly realized that a number of games (such as certain installments of the Final Fantasy series) were only available in Japan. To make a long story short, I created my company, Demiforce, originally as a hacking group in that scene, translating games unofficially to English. This required dealing with games on a machine code level. Without access to original source code, we would write tools to extract Japanese scripts, then translate those scripts, reinsert them, then change the game's code (again in machine code) to display the fonts, kerns, and leadings in the correct order. We had some relatively high profile releases, and it was fun getting a bit of press. But most importantly, it gave me an incredible amount of game development experience, albeit from the inside out. Looking back it was painstaking work but I'm very proud of it and it's definitely helped shape the way I view games, development, and handling press. When Trism came out I figured I'd incorporate Demiforce as a company, and the rest is history.

Why did you decide to launch this?

I had been waiting to make a game like this for more than a decade. I had always wanted to make an indie hit without support from the establishment, to prove to myself that I could do it. However, I always ran into snags. I tried making a game for the Gameboy, and finished it, but couldn't find publishing. That was tough; I didn't play games for years after that. A while later, I tried doing a game for Windows Mobile 6, but found there was just no way to get noticed in the ecosystem. It wasn't until I saw the iPhone that I felt all the pieces were there -- it was a capable device, it had novel features that made it fun to use, and people liked to use it. The only component missing was a legitimate sales channel, so I decided to make it for the unofficial jailbreak community. As luck would have it, it just so happened that shortly after I began, I started to hear rumors of an official Application Store from Apple. Hoping to get ahead of the bell, I did my first ""press release" -- a hastily constructed YouTube video showing off what I had at the time, with a voiceover talking about who I was, when it was going to be launched, and so on. The video went viral, and Apple announced the AppStore the very next week. It hit at exactly the right time, providing enough buzz from gaming sites like Kotaku and Apple sites like TUAW to keep the game fresh in peoples' heads till launch. Trism came out at the launch of the AppStore in 8th place.

If you started Trism while you were working, how did you find the time to do both, was it difficult?

I had been writing barebones, "ghetto" software for a long time, so I was used to working under the gun with less than optimal tools. A lot of the game was created in the back room at my day job, talking with my buddy Robert. Not that I advocate this of course, but I went home early a lot of the time to make sure I could get Trism in. If you have a full time job and are looking to do what I did, man I really don't know. If I had stricter employers it really would have been a struggle. The only advice I suppose I could give would be to avoid stressing yourself out -- I got burnt out 3 times from working too late and paid for it dearly in terms of lost hours or days.



Early Trism Cartoon Splash Art

What is your business model?

My business model was discovered by simply asking my mailing list and forum users what they thought I should charge for the game. Remember, no one knew what an app should look like or an app should cost. I suppose this really doesn't have relevance nowadays.

So when you launched, how long was it until you saw your first sale?

The game sold well straight from the get go -- the first day the AppStore opened, Trism was in 8th place. Traffic was generated by the fact that The AppStore was new, my game was new, and that people wanted to experience a new generation of gaming. I leveraged social media and viral marketing 100% -- I have never once bought advertisement for Trism.

Can you walk us through the tools you used to build the initial version of Trism?

Originally, before the appstore and XCode for iOS, I was using bash, gcc, and notepad to write Trism. It was tough, but hey there was nothing else so we didn't care, it was just a few of us hackers trying to figure stuff out. We really had to rely on each other back in those days, but unfortunately it was difficult because Apple didn't want information about internals leaking, and they were known to shut down sites if they felt it'd be a threat to them. So sometimes we would have radio silence for a week or two where if you had a question, you better figure out how to answer it yourself, or you're screwed.

I spent four months thinking up the idea for Trism and four months developing and drawing it.

What parts of development do you do and what parts do you hire out.

I did all the graphics, all the game design, and 90% of the art for Trism 1. Trism 2 I have professional artists working with me, as well as a server coder (my friend Robert from Trism 1) and a musician.

How was your growth during the first three days, three weeks of launching?

Unfortunately Apple didn't have its metrics reporting built out for the AppStore until about a month after the AppStore opened, and there was no Flurry or Pinch Media to speak of.

What has been your biggest lesson so far?

I look back now and I knew that it was no mistake that I was not working in the game industry. In those days I would always say "someday I'll follow my dream and become a game developer", but truthfully I had crafted my reality for myself. I was never going to get there - I wouldn't allow it. I would try to make games here and there, never thinking they would really go anywhere. I went about it with the attitude of "well, I'll make this game and release it, but if it doesn't work out, it's not my main thing in life, so who cares".

But of course, Trism did hit, and now I am doing it fulltime. But it was not that easy! I resisted at first. I didn't want to leave my day job. One time a woman walked by me in the hallway and said, "I saw you on CNN this morning, what are you still doing here?" and I didn't really have a good answer for her, or for myself. Eventually, I did quit my day job (reluctantly) and started on the path of my destiny. It wasn't until years later, after a lot of soul searching, that I finally decided to become a game developer and truly put my heart into it. And at that point I realized that I had avoided it because I was afraid of putting my heart on the line, I was afraid about being passionate, that I might fail and be hurt. But I know now that is such a small piece of the puzzle that it's totally worth going for your dreams anyhow. I could not have told myself to quit my day job, and when I left I saw many coworkers still trapped in their own prisons as well, but I knew they had to figure it out for themselves in their own time.

What tracking tools do you use?

I use my own custom built analytic engine.

How is your project doing now, and what are your plans for the future?

It's done well, but honestly I am working more towards Trism 2. It's taken us two years to create, is almost ready for release, and we have high hopes for it. Ultimately, I resisted using my fame to catapult Trism into a cash cow franchise -- I didn't feel that would have been very polite to the fans. Instead, I went back to the drawing board and completely reinvented it, focusing on making it something I could look back on at the end of my life and truly be proud of as a body of work.

Domainnr



Idea: Create an app to explore the entire domain name universe beyond the usual .com, .net and .org.

Project Maker(s): Eric Case

Website: <http://domai.nr/>

Browser

How did you get started in building projects?

Oh gosh, I've been a tech nerd for most of my life. My Dad got us an Apple II way back in the 80s and I've been hooked ever since — I remember having a Texas Instruments calculator hooked up to the television, programmable calculators in high school, and getting a Mac in middle school. I started fiddling with web stuff in the early 90s, and ran up large long-distance bills for my parents because we didn't have local Internet dialup in my southeastern Ohio hometown. I graduated from high school in 1997 and was able to do a few web projects in college.

After high school and college, you worked for a while at Google, why did you leave?

I left Google early 2008 after being there for five years. I knew I wanted to start a company so I basically hung out with various hacker friends to see who else might be game; chatting about different ideas to see what might be worth working on.

What did you come up with?

Randy and I were hanging out one night watching *Battlestar Galactica*, and for some reason domain hacks (like "del.icio.us") came up in conversation. It occurred to me that we could snag the full list of top-level domains from Wikipedia, and prototype something in JS to show, in real-time, what domain hacks would be possible with them. Randy coded up a demo on my Mac Mini that night, and it named itself "Domai.nr" as we fiddled with it. We tossed it on a web server and both kept using it, while we were brainstorming various product ideas to possibly work on. So the short answer is, we liked domain hacks and happened on a way to make them easier to create.

First version, Domai.nr April 2008

Many projects take some time to find their product/market fit. Why do you think Domai.nr has been such a hit from day one?

It's probably some combination of:

- 1) The awesome UX Randy created.
- 2) Cleverly named, (SEO-friendly and memorable).
- 3) Fast and stable.

A more general take on this is our values—that we expressed in the product: it should be fast, easy to use, have minimal UI elements, no evilness, be developer-friendly, and get out of your way a la Google in the old days. The UX is also incredibly sticky. According to Google Analytics, average time on site is 10-11 minutes.

How did you overcome building your project in the face of existing competition?

We weren't aware of anybody doing what we did with Domainr, which was putting the entire top-level domain namespace in a single search box. Most domain search sites were run by Registrars, and only included top-level domains that they registered. They were approaching the problem from the position of their own product offering, while we approached it as a meta-search problem for our own needs — brainstorming tools for products and hacking.

Was it just you? Did you do the design, coding, marketing, all by yourself?

I had two partners, Randy Reddig and Cameron Walters, who are now both co-founding engineers at Square. They are the coders; I do our operations and the business side of things. Funnily enough, Randy and I “knew” each other online since the mid-90s via a map-making community for Marathon (a Mac game from Bungie Software, which was later acquired by Microsoft and then went on to make Halo). We didn't meet in-person until probably 2007 or so, when he was at Six Apart and I was at Google. I knew Cameron from the general web/tech scene here in the Bay Area. It's a relatively small community, I think.

How did you know this would be a worthy project, and how did you get started?

Well once Cameron joined us we had enough manpower to actually build it out, and it was our best idea that a) we continually used ourselves [thanks to the original JavaScript prototype], and b) could generate revenue via affiliate relationships. It followed the principle of building something for yourself and see if anybody else has that same need. And then see if it can earn money.” We knew we needed some time away from our daily routines to focus on it, so we rented a house up in Mendocino for a week with the goal of finishing it. During that week Cameron built out the backend, Randy redesigned the frontend, and I did everything else — gathering and normalizing the data via Mechanical Turk, testing, affiliate setups, etc. We launched it the last night we were there.

So you decided to build it, were there any initial costs?

The domain itself (domai.nr) was \$500, which was initially our biggest financial curiosity — “let's see if this thing can earn enough to cover its costs each year.” Our SliceHost server was around \$20/month, and that's pretty much it. The other costs were our time, and any material things we did to support development—the Mendocino road trip, meals, etc. We worked out of our apartments for a while too.

Total startup cost: \$520

What kind of software and tools did you use?

The frontend currently runs on Google's App Engine. The backend is hosted at SliceHost through which we proxy the DNS requests using Twisted. The app we use to manage all the domain and registrar data is a simple Rails app, and we host the source on Github. I use Google Docs for most of my operational and analytics work, to keep track of Domainr's business side.

Did you have to learn anything new for this?

Randy and Cameron hadn't used Python or Django before, which they learned in order for us to use with App Engine. Same with Twisted. I hadn't used Mechanical Turk before, which was pretty nifty.

What did you use Mechanical Turk for; can you explain how you used it to normalize data for the app?

The domain namespace is chaotic—every top-level domain and registry seems to have its own rules for things like minimum character lengths for domains, whether or not you can register at the top level (foo.nr

vs. foo.com.nr for example), and I didn't want to go compile all these nuances by hand. So I used Mechanical Turk to gather things like the min-char lengths for each top-level domain, top-level registration possibilities, and all the second-level domains they may or may not use (Brazil is the craziest). I went through and cleaned up the results when the Mechanical Turk job was done, and that was our initial dataset. We've tweaked it ever since, but that core is what powers Domainr's search logic today.

How did you deal with technical issues?

We pretty much searched Google for everything we needed to know. Randy and Cameron are seasoned engineers — Cameron has his Masters in CS, and Randy has been coding since the mid-90s.

How long did it take to get your first version done and up for sale, or MVP?

A few days. We already had a working JavaScript prototype so we were basically building it out from there.

After building, how quickly until you saw your first affiliate sale?

They started happening immediately via the popular registrars—GoDaddy, Network Solutions, Dotster, etc.

That must have been a good feeling, how did you continue marketing it after that?

It's been word of mouth all along, primarily via Twitter. We've gotten some great coverage on various design and tech blogs over the years, and also tried to make it an SEO-friendly site. The short domain craze has probably been our biggest driver of growth, as many people with existing sites are registering new short domains to share links on twitter with. We couldn't be happier about this.

How was your growth during the first three weeks of launching?

There was a massive spike when we launched, then traffic immediately dropped to like, none. We didn't get another spike until ~four months later, and growth has been very gradual ever since then.

Tell us about how your iPhone app came to be.

Sahil Desai emailed us out of the blue one day to tell us how much he loved Domainr, and that he wished we had an iOS app for it. He offered to build one for us, just for fun. We were totally stoked by this, met up with him for coffee, and ended up building Domainr's API to support his work. It took a while to get done because we were all busy with our day jobs and whatnot, but eventually we shipped it. We've stayed in touch with him ever since — he actually spent a bit of time contracting at Square before joining Bump, as a result of us working together.

How is Domai.nr doing now and what is your future plans?

It's continuing to grow slowly, and our plans are to keep working on it in side-project capacity as it's basically paying our respective apartments' rent each month. We're in the middle of porting it off App Engine over to Rails, the stack with which Randy and Cameron are most familiar these days. Once that's done, we've got some more frontend iteration we want to do, to see about making the site even more useful.

What advice would you give to someone wanting to build their own side project?

Speaking for myself here I like working on projects that solve actual problems I have. I like passive income streams. I like simple solutions to problems. I'd urge people to think about innovative ways to solve the inefficiencies and problems they see in their own lives, and the wider world. The world is full of things that should be better.

Monkey Bongo, Enigmo



iPhone

Idea: Create an iPhone version of our hit game, Enigmo, for Mac.

Project Maker(s): Brian Greenstone

Download App: <http://itunes.apple.com/us/app/enigmo/id281736535>

<http://itunes.apple.com/us/app/monkey-bongo/id459431520>



Tell us a little about how you got started in web development and software.

Actually, I've yet to get into Web development – I know basic HTML, but that's about it. However, for game development... I've been doing game development since I was in Junior High. I got an early start writing games for the Apple II+, and in 1984 I almost got my first game published by Broderbund, but the deal fell through. It wasn't until 1988 that my first game got published, and that lead to a full-time career making games for the Playstation, Nintendo, Mac, and now iOS devices.

What were you doing before you built Enigmo?

To be honest I was trying to figure out what my next career move was going to be. I had been doing games for a long time, but I was getting really burned out. Then the iPhone came along and we took Enigmo which was originally a Mac game and it suddenly became a huge iPhone hit. That got me interested in making games again, so it's been a crazy 3 ½ years.

What was your big differentiator?

Apple gave the iPhone development kit to developers in March of 2008, and we knew that we only had until July to get things done. Enigmo was the simplest game we had ever done for the Mac, so it seemed logical to try porting that to the iPhone first since it would be easy. We didn't realize just how good the game would be on the iPhone, but it ended up being a huge hit. When Apple asked me to present at the 2008 WWDC keynote with Steve Jobs we were originally thinking that we'd be showing just Cro-Mag Rally, but Apple liked both games and wanted me to demo them both. That was a good thing because both have sold over 1 million units each.

Were you working when you started to build Enigmo?

I wasn't doing anything when the App Store came along. I really was burned out and had started doing some photo utilities to get out of gaming. But once I got my hands on the SDK I got busier than I had been ever in my career. Working ridiculous hours to get the entire Pangea game library ported to the iPhone. The beauty of it is that the cost was \$0. We already had the games, and almost no new assets were needed to do the iPhone versions. It was all just about modifying the code, and most of the games were up and running in 2 days. Almost all of them were done in 2 weeks.

What is your business model?

I've been very happy with the way I've set things up. I am the only actual employee of Pangea Software. Everyone else is a contractor. Typically I just pay royalties to them on the games – that way it costs nothing to get the games developed. No risk. In the end, however, I almost always end up paying more than had I just paid them for their work, but when they make tons of money everyone is happy. Also, not having employees means less paperwork, less taxes, and more time off in between projects. Everyone works out of their homes, and we trade files back and forth via email or iChat. It's all very efficient.

How long was it until you saw your first sale?

Pangea Software was formed especially for a game I did called "Xenicode". It was for the Apple IIgs back in 1987, and we made money on it almost immediately. We did not publish that ourselves – another company did that for us, and we didn't make a ton of money, but for a starving college kid it was great! Once again, our development costs were \$0 on that.

Social Media and such didn't exist back in the 80's. We had some following on the old AppleLink PE which later became America Online, but in those days it was traditional media that got you exposure. We had to get reviews in all of the big game and computer magazines.

Can you walk us through the tools you used to build the initial version of Enigmo?

The original Mac version of Enigmo took 6 weeks to make. I had the idea for the game since college, so I knew what I wanted to do. It was really a simple game – not a lot of art or sound assets, so once the basic engine was working the game just fell into place. It was made with CodeWarrior, Photoshop, and the 3D modeler called Form-Z.

What parts of development do you do and what parts do you hire out?

I do all of the programming and marketing. I do all of the coding, so if a bug comes up I fix it. No biggie. I also do the sound effects. I hire out all of the art, animation, and music development to a set of guys who I've been working with for a very long time. On the bigger projects like Bugdom 2 and Otto Matic I had to pay them some royalty advances so they could eat for 10 months, but on the smaller projects everyone worked just for royalties.

How are you getting users for the product?

Pangea has been around long enough (24 years) that we've got an established fan base. We have a Twitter following and we also have a messaging system in our games to let users know when something new comes out. We also send out a lot of Press Releases and hope that Apple will show us some love whenever we have something good coming out.

Prior to the iPhone, Pangea's best year made about \$1 million. The first 7 years of the company's existence made around \$15k. After 1999 we made less than \$1m per year, but the iPhone change that when in 2008 we did \$4m in sales just from July to December.

How was your company's growth been?

No growth really... not in 24 years of doing business. I still run the company out of a spare bedroom, I still hire only contractors, and I'm still sitting in the same office chair that I bought in 1993.

What has been your biggest lesson so far?

The most profitable games are not always the best games. Enigmo is the perfect example. I had always considered the Mac version on of our B products, but lo-and-behold it was a fantastic mobile app. Then you've got a game like Otto Matic which in my opinion is the best game I've ever done, and it took a year to make, yet it flopped on the Mac and the iPhone. Go figure.

What tracking tools do you use?

I stopped caring about that stuff a few years ago. I found the information to be not very helpful and somewhat repetitive. I know who my customers are based on the emails I get, so I know who we're targeting for future products.

How is your project doing now, and what are your plans for the future?

We're almost done with "Monkey Bongo". Should be out next month!

What advice would you give to a developer that is just about to begin a side project?

The key to success in the mobile marketplace is to crank out as many good apps as you can as quickly as possible. Most apps will flop, but you're aiming for that one surprise hit like Angry Birds. It's far easier to make a "lot of money" by making a "little money" on lots of individual apps.

BatteryBar



Browser

Idea: Build an app that monitors the status of your battery and displays your battery's status in the taskbar.

Project Maker(s): Chris Thompson

Website: <http://osirisdevelopment.com/BatteryBar/purchase.html>

When did you start the project and come up with the idea?
I started writing BatteryBar around August 2008. The idea started with a comment from my boss at my day job who asked me, "How do I get that battery meter to show?" after getting a new laptop. I knew he was referring to the Windows battery icon, but it sparked the idea that people should be able to see the time remaining right on the taskbar without having to hover over the battery icon. That weekend I started on the first version of BatteryBar.



Were you employed when you decided to take on the project?

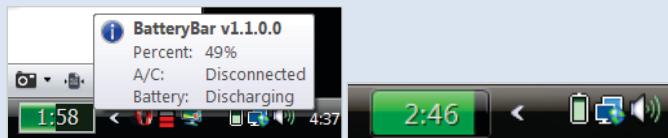
I was employed full-time at the time and worked on the project on evenings and weekends. I'm still employed by the same company for my day job.

Had you built any other side projects before this?

My first real project was a desire that I had to build a website for my family to be able to communicate, post messages, pictures, and current contact information. This was back around 2000. I taught myself PHP and MySQL and over time built out a website that we continue to use today. I had plans to build it into a platform that would allow other families to build their own sites, but I never put in the time to finish that project. For work I had done several projects that I built on my own because of a need, not because I was asked to build the tool. One of them involved tracking jobs that needed to be monitored from start to finish with detailed billing information for a client in Palo Alto. The second was a program to batch print PDF and TIFF files, which was a common task our employees would have to do. BatteryBar was the first tool I built that I generated a revenue stream for me.

What was your original goal with the project? Did you think you think it would make money?

The first several versions of BatteryBar were freeware. My main goals were to build a product that displayed the time remaining on the taskbar and that would provide a battery time estimate than Windows. The second goal really drove the development to differentiate BatteryBar from other laptop battery meters. Originally I didn't think about selling the program, but as I researched other similar programs I found that most had gone defunct or hadn't been updated in years and most sold for around \$10. It was then that I figured I'd turn the product into a freemium product and sell advanced features while still providing the core functions for free.



How did you get started after your first thoughts on the project?

First, I determined exactly what I wanted to do. The goal was pretty straightforward: 1) It had to display on the taskbar, and 2) it had to collect data from the battery. From there I researched the iDeskBand

Windows interface and started writing sample toolbars until I could figure that out. I also researched the Windows APIs needed to query the battery. There were some basic samples online on how to do that. The first version collected data a lot like Windows, but averaged the last 3-5 samples to smooth out the time remaining estimate. A month or two later I scraped that system to go with a completely custom solution to calculate time remaining.

Was it difficult finding time to build since you had a job? When did you work on it?

I found that if I had a goal I needed to accomplish, I could get pretty absorbed into writing code, or researching how to accomplish something, for hours at a time. I spent my entire commute time on the BART traveling to San Francisco for 40 minutes each way working on the program. I also spent lots of time in the evenings and weekends. Development slowed only because I didn't know what to do next or what new feature to add. As long as I had a plan I was able to really focus on it. At first I was working on a free version, so I worked on it out of a desire to learn and improve my skills more than to earn revenue.

Did you work with a partner?

It was just me. I did everything from writing the software to building the website and integrating with the various payment processors I've used. After I had a large number of users I used their input and suggestions to determine what new features I should add. That continues to be the model today. The users really come up with all the feature ideas.

What tools did you choose to use to build it?

BatteryBar is written in C# using the .NET Framework 2.0. At the time my development experience was mostly PHP/MySQL and VB.Net. I used this as an opportunity to teach myself C#. I knew early on that C/C++ would have been the best platform to build an explorer extension due to limitations with .NET as a COM object, but I didn't have any background in C/C++ and used it as a learning experience for C# and P/Invoke. I focused on Windows because that's what I had access to as a development platform. I didn't have access to a Mac and had no experience writing software for Linux.

What new things did you have to learn for this project?

The biggest things I had to learn were C#, how to develop a COM object in .NET, and how to use P/Invoke to make low-level Win32 system calls.

How did you deal with technical issues?

There were several technical hurdles that I had to deal with, primarily with issues with the iDeskBand interface. I spent a lot of time on Google searching for any code samples or comments I could find about the issues and testing code samples. I spent a lot of time tweaking code and running it through the debugger.

How long did it take to get your first real version (MVP) up and running?

It took about a week or two to get the very first version working. I mostly tested it on my own laptop. I was able to make sure that software worked on my laptop. I put together a simple website and starting posting it to download sites. From the users I got I started learning about quirks and differences with other laptops. Those early users helped me polish the software and provided a lot of great feedback. Several would run debug builds that would log data for me to help improve the code.

Did you take care of the other aspects of the project, like the web design?

I did everything on the cheap. I used a hosting account I already had for my family website and found a free template online. I built a static HTML site and just updated it when new releases would come out. Since then I've moved on to a VPS but the cost is quite reasonable. The only real costs besides my own time were marketing expenses.

What kind of business model did you decide to go with?

I decided on a freemium model. I made that decision primarily because I had already released the first 4 versions (v1.1, v2.0, v2.1, v2.2) as freeware and I didn't feel it was right to stop the free version altogether. I had also found that the free version spread quite quickly online with a lot of people posting reviews on blogs or in website forums. The word of mouth of the free product was and still is the biggest

marketing tool. Had I not gone with the freemium model, I don't think I'd have the number of users that I have now.

I started off by selling the Pro version for \$7.50. At first there weren't a lot of features that were only in the Pro version, but over time all the new features were added to the Pro version with only bug fixes and some minor updates showing up in the free version (which is the same software with a code to unlock the Pro features).

Since then I've played around with a lot of different pricing options, from Name Your Own Price (like the Humble Bundle) to different prices for time-limited licenses, to a single price.

After building, how quickly until you saw your first customer or sale?

My case is a bit unique. I already had a free version in the wild with quite a few users. I released an automatic update to users using the current version (2.2) to upgrade to 3.0, which was the first version that a licensing system. I saw sales on the first day from current users that upgraded and decided to buy the Pro version. In the first month I made about \$400 in sales.

How did you continue marketing it after that?

Most of my marketing was from word of mouth. In the second month (February 2009), I was lucky enough to have LifeHacker post an article about BatteryBar (<http://lifehacker.com/5145063/batterybar-adds-a-power-meter-to-the-taskbar>). You can still see their screenshot of version 3.0, my first freemium release. That month sales jumped from \$400 in January to about \$1,800.

After that I made a lot of effort to search Google and find where people were posting about BatteryBar and responding to the posts or threads. I also emailed a lot of blogs trying to get them to review the software. Unfortunately most of my efforts didn't result in a lot of traffic. It also all came from users that posted or recommended it.

I have over the last few years used AdWords to advertise. It's always brought in a lot of traffic, but it's difficult to determine how many people upgraded the free download the Pro version because of the advertising. Despite that, I've still put a few thousand dollars into ads, usually less than \$100 a month, and some months I didn't do any.

Right now I think that the best marketing tool would be to get a professionally designed website and improved shopping cart that would increase conversions. With that in place, I think the ROI on the advertising would go up.

I saw that you had a feature in life hacker, did you get in touch with them or did they discover your project?

Honestly, I don't know how they found me. According to the blog post, they site Download Squad as the source who sites TechnoSpot, who sites Technixupdate.

I think that those blogs discovered my software because I was posting and uploading the program to every download website I could find. I made a PAD file and uploaded that to dozens of sites a day, searching endlessly for every freeware/shareware download site that would take submissions. My guess is that someone saw BatteryBar on one of those sites as a new submission and decided to review it.

It was a big break. Because of that I tried to contact a lot of other big and small blogs but was only featured on a few. I have been featured in a number of other websites including some well-known ones, but they did that on their own. I've also been featured on some magazines in the CD-ROMS and even a front-page on a Polish computer magazine that paid to have a custom Pro version to give to all their readers.

Did you submit your software to CNET?

I did. You can find BatteryBar on CNET's download.com (http://download.cnet.com/BatteryBar/3000-2094_4-10866804.html) and many other sites. As I mentioned earlier, when I first released the Pro

version I submitted a PAD file to every download site I could find. I think that was probably my most productive marketing technique.

How did customers respond to your business model?

The free version is quite popular. I estimate there are a few hundred thousand active users around the world based on the number of automatic update requests the server receives every day. At first, there were a few angry customers because I moved two features into the Pro version. I provided them with free license keys. Since then I've found that about 1-2% of users will upgrade from free to Pro. The free version provides the basic functionality that most people want, including my custom time remaining estimation code.

I've had people upgrade to Pro before even installing the free version and I've had people that have been running the free version for years who will upgrade. Unfortunately all the evidence I have is anecdotal; I don't have any real way of determining what free users convert since I don't ask for any personal information to install the free version.

How was your growth during the first three weeks of launching?

If we go way back to the first version, I'd say that growth was small but steady. Most people learned about it because of all the download sites I was posting it to. I wasn't doing advertising at that time. I would also actively Google "BatteryBar" looking for anyone that was posting about it and trying to be active in the discussion. Within the first few weeks I probably had less than a thousand downloads.

My first Google Analytics data starts on October 10, 2008. In the three weeks I had about 1,700 downloads or so.

What about the first 3 months?

The first three months of data I have show approximately 10,000 downloads from October 10 to December 31, 2008. That was with the free version.

The first three months of the release of the Pro version shows about 70,000 downloads, with about 45,000 of those coming from referrals from LifeHacker, Download Squad, and a thousand other referring sites. The Pro version had 440 sales totaling \$3,200 gross (\$2,700 net after processing fees).

Is BatteryBar your full time project now?

No it is not. At this point the product is fairly mature and I don't spend as much time on it as I used to. I spent probably an hour or so a day on support. I spent more time when I come up with an idea for a new feature or identify a bug that needs to be fixed.

**Any other stats you would be willing to share about the product from the beginning until now.
(Total revenue, users, downloads, etc.)**

Due to switching payment processors a few times, I have a hard time determining the total revenue for the past two and half years. I have sold approximately 12,500 Pro licenses during that time. I have also had several enterprise customers and special deals during that time frame that adds an additional two or three thousand users.

Do you consider yourself more of a designer or developer?

I consider myself a developer. I really wish I had a better design sense.

What are some of your other side projects? How did you come up with those ideas?

Before BatteryBar I actually had another side project that I was trying to make into a commercialized product called AutoUSB. The idea was to automatically run a backup whenever a certain USB device was connected. I came up with the idea because my Mom wanted to keep backups of her laptop but all backup programs I could find would only run on a schedule and we could never guarantee that the USB drive would be connected at a certain time. I had a working version that was nearly done, but I hit a snag and I let it convince me to stop working on it. To this day I wish I had finished the project. I've had several other ideas for projects but haven't been able to make them happen. My latest idea has to do with online

photo backup and sharing. I just don't have all the expertise that's needed to get it written and working and I've had a hard time finding someone with the right expertise that'd be willing to work with me on it.

Do you have any guidelines or rules for coming up with ideas for side projects?

Ideas are the easy part. Just look around at anything that bothers you, or better, bothers a friend or family member that's less computer savvy.

What advice would you give to someone wanting to build their own side project?

My advice would be to find the smallest set of requirements/features necessary to get something out the door. BatteryBar 1.0 was enough for me to get it in front of other people, but was really very simple. Just releasing it and getting out there created interest and the users pushed for more features. If you make a grand list of everything you want it to do you'll get overwhelmed and you won't get anything released. Start small and incrementally make it better.

How your app doing now and what is your future plans?

I make on average about \$50-60 a day in sales. I wish it was more and I plan on finding someone to design a new website that I hope will increase conversions. The biggest areas I need help in are design (especially the web) and marketing.

My future plans involve creating an enterprise version of the product that I can sell that will include a client and server component that will report statistics of a laptop or any computer back to a server so that an IT department can be informed when batteries need to be replaced or potentially other issues with the computers.

I also have several ideas that I'd love to work on, but I lack technical and design expertise to get them off the ground. I really have a desire to found a startup, perhaps with one of the ideas I've mentioned or another. My day job and activities don't put me in contact with many developers, so I've found it hard to find someone to work with.

I'd love to be a co-founder on either a project idea I've had or someone else's. If you're interested in talking to me, I can be contacted at chris@osirisdevelopment.com.

Fantasy Monster



iPhone

Idea: Create a fantasy football iPhone App.

Project Maker(s): Jerry Shen

Download App: <http://itunes.apple.com/us/app/fantasymonster/id371000743>

So what were you doing before you built Fantasy Monster?

I was a software engineer at a defense firm. Definitely a cog in the machine (150K employees).

Do you consider yourself a designer or developer?

A bit of both. I'm definitely a software engineer by training, but I have an interest in UI and UX. I lack the Photoshop-fu of a true graphic artist so I outsource the final design but I'm definitely one of those developers that feel more comfortable making GUIs than command line tools.

How did you come up with the idea?

I've always been a pretty avid fantasy sports player since I was a kid. One of my final classes during my master's program was an iPhone development class. I needed an idea for the final project and fantasy sports seemed like a good fit. I did a bit of research on existing apps and it turned out they all pretty much sucked, so I thought there was a definitely opportunity to disrupt the market.

What are some of your other side projects? How did you come up with those ideas?

To be honest I'm not really a side-project kind of guy. I've always liked having a balance in life. If I'm sitting at my desk programming during the day, the last thing I want to do is hack on some side project by night. That said, fantasy monster was technically a side project, but it started off as a course assignment. Had I not been forced to do it for my master's class, I likely wouldn't have done it.

Weren't there already fantasy football apps out there? Why was this still worth doing for you?

There were, but none of them were good enough for me. I set out to build something that I would find useful even if no one else bought it.

How did you get started?

I read Beginning iPhone Development by Jeff LaMarche. It was great because it had hands on style much like Learn Python the Hard Way (Zedshaw's Python book). After a few chapters I was comfortable enough to at least start wire-framing the UI in code.

Did you do the design, coding, marketing, all by yourself?

For the first app I did everything. Now I've hired a part time designer and Android developer.

Did you know anything about iPhone development?

I didn't know anything about Objective-C or Cocoa prior to starting. Honestly, I didn't hit any major technical roadblocks. The development went pretty smoothly. Most of my initial hurdles were on the business end.

How did you deal with those?

For tech hurdles I used Stack Overflow pretty extensively. I'm not the smartest developer so having Stack Overflow probably doubles my productivity.

The screenshot shows the Fantasy Monster app interface. At the top, there's a banner with the app's logo and a link to the App Store. Below the banner, there are three main sections: 1) 'Auto Draft' showing a progress bar for a draft. 2) 'Check-in On Your Leagues' showing a progress bar for checking in across multiple leagues. 3) 'Manage Your Fantasy Teams' showing a list of teams with their names and logos. The overall theme is green and white, with a clean, modern design.

How long did it take to get your first version done and up for sale, or MVP?

It took me 5 months working part time from writing the first line of code to submitting to the app store.

After submitting to the iPhone store, how quickly until you saw your first sale?

I had a sale almost immediately. Almost everyone who submits an app has at least one sale the first day because you get visibility from the new releases list.

How did you start or continue marketing it after that?

Originally I contacted a lot of iPhone review sites, and that was pretty much a bust. I eventually realized that my customers probably weren't geeks who read iPhone review sites anyway. They were more likely to be in a bar picking up chicks and arguing about the super bowl, so I started to contact sports-specific blogs instead. I got a limited reception there as well. Eventually, I sort of gave up on "marketing" and just worked on refining my product and adding more features (like support for different leagues).

How was your growth during the first three weeks of launching?

Not very good. I averaged maybe \$30/day? The people that bought my app loved it but not that many people were biting.

What about the first 3 months?

Things picked up significantly in mid-August 2011, when I released a football specific version of the app. This was basically 3 months in.

Can you share any numbers during that period (users/monthly net)?

Yeah first 3 months I made \$9,916.

The next 3 months I made \$65,881. So there was definitely a huge inflection point 3 months in.

How your app doing now and what is are your future plans?

The app is doing great. I've ported the app to android and iPad. I've also got a 2nd app out (DraftMonster) on iPhone that I'm planning to port to iPad and android as well. My revenues have basically tripled compared to last year due to the fact that I've got more apps and I support more platforms now. My future plans are to continue improving the product and listening to customers. It sounds simple but I really think it is. Do right by your customers and they'll do the marketing for you. When my second app was released I was nowhere to be found on the new releases list due to an app store glitch, but it still ended up getting into the Top 5 sports apps on iPhone just purely based on downloads from existing customers.

Bingo Card Creator



Idea: Create an app that helps teachers create bingo cards efficiently.
Project Maker(s): Patrick McKenzie
Website: <http://www.bingocardcreator.com/>

Browser

You are a legend on HackerNews and other forums as “patio11,” most people know you as “patio11” there.

Tell us about Patrick McKenzie, and how you got started.

I graduated from Washington University in St. Louis with degrees in Computer Science and East Asian Studies. To make a long story short, I really wanted to be an engineer, but was worried about increasing competition from China/India, so I thought if I did one really hard thing plus engineering I would have a nice safe job at Microsoft for the rest of my life. Japanese is really hard, and Japan spends billions on American software, so I thought that was a natural fit. After graduating I went to Japan to firm up my business Japanese.

My first job was as a technical translator at the prefectural technology incubator here in Gifu. (I won't tell you which one, but since Gifu is Japan's answer to Kansas, that narrows it down quite a bit.) I was a bit under challenged at my job, so I spent quite a bit of time playing World of Warcraft in the evenings and wondering "When does my career, you know, actually start?" Eventually I decided to take matters into my own hands and create Bingo Card Creator, a small software business. I had no idea whether it would be successful and was aiming mostly to learn things about selling software on the Internet. This would have been back in late June 2006. BCC has grown gradually since then. I eventually switched jobs from translator to engineer and spent a very hard 2.5 years at a Japanese megacorp before reassessing what I wanted from life again. In the interim, BCC had grown to the point where it would support me, so I went full-time on it as of April 2010. Since then I've been doing BCC, consulting, and working on my new business, Appointment Reminder (of which more later).

You started Bingo Card Creator while you were working. How did you find time for both?

Bingo Card Creator was launched on a budget of 8 days and \$60, and has been profitable since. The main reason that I had time to do both in the beginning was that my job was quite cushy indeed with regards to hours – somewhere in the 35 to 40 region. By the time I became a salaryman and was working 50 to 70+ hours a week, I had already optimized much of the running of BCC, such that I could e.g. throw a pair of A/B tests up in the weekend and just answer emails briefly during the week.

I'm very happy that I started part-time, since it reduced the stress of running a business considerably. In this last year, I've never been close to starving, but I definitely have transient money issues – e.g. can I afford to buy my ticket for Christmas now, or do I have to wait until October when BCC has a big month? With a full-time paycheck, that never even entered the radar screen. Also, with my general level of risk tolerance, I probably would never have been comfortable taking the leap without first testing out that I was able to swim.

So when you launched, how long was it until you saw your first sale, and where did it come from?

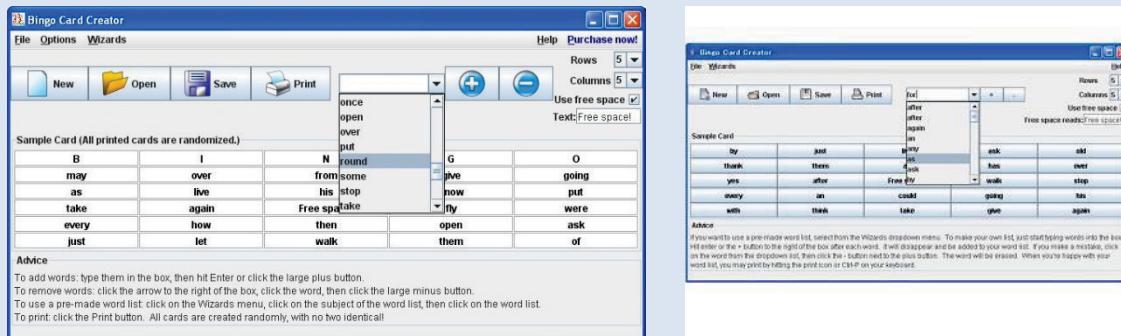
BCC had its first sale approximately two weeks after launch, to a searcher who found an article I wrote about [Dolch sight words bingo]. If you were an English teacher, you'd have heard of the Dolch sight words, which a pedagogist identified many years ago as key to early readers' understanding of English.

A screenshot of the Bingo Card Creator website. The header says "Bingo Card Creator". Below it, a large image shows a bingo card with the word "BINGO" in the center and various words like "give", "name", "old", etc. around it. The page has sections for "Create Your Own Bingo Cards Now!", "Featured Bingo Activities", and "Less Work, More Fun and Educational Bingo Games!". There are also sections for "Featured Printables Categories" and "Save time! Save money!".

There are about two hundred of the words, grouped into five grade levels. Many teachers don't know which words are on which grade's list, but they know enough to search for them. I figured that if I put up the word lists and provided an activity to go with them (Dolch sight words bingo), that would be an effective method of selling BCC. This was a major success and provided most of my sales for the first year.

Can you walk us through the tools you used to build the initial version of BCC?

The first version of BCC was a downloadable Java Swing program, written in Eclipse. The website was a template from <http://oswd.org> (Open Source Web Design), customized by hand in Notepad, since that was the limit of my web programming ability at the time.



Early Version, Bingo Card Creator

What parts of your side project do you do and what parts do you hire out?

At the start, I did absolutely everything for BCC. Since then, I've largely outsourced web design and writing new content for the website, but continue to do most of the programming, marketing, and customer support etc myself.

What technologies are you using now?

As I grew in programming ability, the BCC marketing site transitioned from a static HTML site to a Ruby on Rails application, largely because this allowed me to conveniently publish large amounts of bingo-related content. (I have 942 or so individual bingo activities, with the intent of ranking for any activity a subject could possibly want to teach a lesson about using bingo.

See <http://www.bingocardcreator.com/bingo-cards/>)

After I had the site running in Rails and figured out how to programmatically generate bingo cards without using the Java code, it was only about another three weeks of work to make an online version of the software. I launched this in summer 2009. Online software is hugely superior to downloadable software, for reasons which I've covered extensively on my blog: <http://www.kalzumeus.com/2009/09/05/desktop-apps-versus-web-apps/>. I've gradually all but eliminated the downloadable version, and most customers now use the web version. Conversion rates to purchase on it are about double what they were on the desktop version, and support costs are literally an order of magnitude less. I'm extraordinarily happy about this result.

Can you walk us through all the different sites that make up Bingo Card Creator?

Bingo Card Creator is at <http://www.bingocardcreator.com>, and is my primary marketing site for BCC, and also hosts the web application proper.

My blog is at <http://www.kalzumeus.com>. It doesn't do much for BCC these days, but was instrumental in getting links in the early years.

I have a stable of micro-sites covering individual bingo activities. Here's two which are fairly representative:

<http://www.halloweeningocards.net>
<http://www.christmasbingocards.com>

These function mostly to pick up searches for their one activity and funnel a portion of those searchers into the trial for BCC.

Micro-sites, for someone who doesn't know SEO too well, would you recommend this approach to them? Did it take long to show up in search results and start getting customers from it?

The reason those micro-sites work is that they are on exact match domain names. SEOs call those EMDs. If, for example, you own halloweeningocards.{com,net,org} and someone searches for *exactly* [halloween bingo cards], you get a major bonus to your ability to rank for that particular search.

EMDs make up a fairly minor portion of my overall income for BCC these days. I wouldn't call them core to my strategy, but they're quite profitable. Whether I would recommend you to try them for your business depends on the particulars – if for some reason there is an EMD on a high-volume keyword which is clearly relevant to you that is available for \$8.95, absolutely, go to town. Would I suggest buying one for \$20k if you were just going to put a micro-site on it? Probably not. Then again, with the cost of AdWords in some commercial verticals, I could think of EMDs in enterprise software markets which would be a *steal* at \$20k.

While building Bingo Card Creator, you built this side project from scratch in a week and blogged about it. It was really motivating for people to read about your success with the project. Do you think the blog itself helped that success at all? What kind of advantages did you have by being so transparent?

I have been blogging since I started back in June 2006. My specific reason for doing that was that a particular blog post by Brian Plexico, who made skeet shooting software, showed me that it was actually possible to run a part-time software business. (<http://www.microisv.com/archives/2006/03/06/conception-to-sales-in-7-days-can-it-be-done/>) Prior to that, I thought it was impossible – that there was some amorphous law of nature which said that if you didn't have 80 hours per week, you should just give up.

Over the years, I've developed a bit of a reputation for saying true and useful things in an amusing manner. That reputation is immensely valuable to me. For one reason, links to my blog improve my ability to rank for bingo-related searches, providing a direction SEO benefit to my website that I wouldn't have if I was just quietly working in my kitchen. For another reason, my Internet presence (blog, HN, various forums, conferences, etc) acts as fairly effective networking for my consulting business. There aren't that many major tech companies here in Ogaki, but folks who have money and problems know my name as a result of my Internet activity, and they trust me to take their money and fix their problems. I'm often happy to oblige.

What do you have to do to "maintain" bingo card creator now

As of writing, I've sold about \$26,000 of BCC this year. I anticipate I'll round out the year at or near \$50,000, modestly up from \$46,000 last year. Profits are generally somewhere in the neighborhood of 60% of sales.

BCC is in maintenance mode right now, so the only thing I routinely do is answer customer support email. I average twenty minutes of it a week. (Prior to quitting the day job, I averaged about 5 hours a week over several years, including development, marketing, support, and the like.)

My hourly wage for BCC in 2011 is probably somewhere north of \$500.

What is the biggest mistake you made while building this project?

I did many things which were in hindsight suboptimal, but I'd like to think they were learning opportunities. I probably could have quit my job earlier if I had had more confidence in my own abilities, particularly as a consultant, since that massively de-risks running a product business. (If you have a bad quarter, sell a week or two of your time and wait for a good quarter.)

What has been your biggest surprise?

Hundreds of thousands of people urgently need bingo cards to teach their kids how to read, and thousands of them are willing to pay \$30 to get them.

Bingo Card Creator is a great side project that I think many developers are aspiring to create something like. You've since started another side project, Appointment Reminder. Let's talk about that one. Can you tell us about it, how you came up with the idea, the target audience, and how it's going? When did you start it? Have you considered live blogging the journey like Bingo Card Creator?

I don't know if I would call Appointment Reminder a side project. My intention is that it is a business and it will eventually comprise the bulk of my income. Sadly, the first nine months of it have not exactly had a rocket ship trajectory, but I'm still working on it.

Appointment Reminder (<http://www.appointmentreminder.org>) was one of a couple of dozen ideas sitting in a notebook when I quit my day job. I had experimented with Twilio a bit and was sure that it was going to revolutionize many businesses, so many of the ideas were Twilio-related. AR was clearly within my capabilities to actually deliver, and pushed buttons for customers I talked to. Plus, when I implemented a MVP of it (a simple faked calendar which, if you clicked to "schedule an appointment", the computer would call you and mark you as confirmed for your "appointment" or "canceled" prior to reading you a prepared spiel about how good of an idea buying AR was), people were ready to write checks.

I released the MVP in approximately May of 2010, got distracted by life for a few months, then officially launched in December. Since then it has been slowly growing organically.

Did you take a similar approach to developing Appointment Reminder as you did Bingo Card Creator?

Appointment Reminder is a Ruby on Rails application, notably using Twilio for phone calls and SMS messages. My initial costs were on the order of \$2,000, for web design, hosting, and Twilio credits. The MVP took about two weeks to write and publish. The actual application took about six weeks from the first day I wrote non-MVP code to the day I started actually charging customers' credit cards.

My goal for Appointment Reminder was to have 200 paying customers by December 2011. Candidly, that looks unlikely at the moment, largely because other priorities have kept me from marketing it the last few months. I finally got back in the saddle in August, and then got sideswiped by Immigration issues. Ahh well. It will happen when it happens.

Long term? There are a few possible futures. If AR sold \$20,000 of service a month, that would leave me sitting pretty happy. It is possible that I'll eventually decide that what I really want to try next is to try running a company which takes investment, and if I ever decide that, I think AR is a good candidate for it.

How are you getting users for the product?

To the surprise of essentially no one, my main strategy for marketing AR is through organic SEO. I am not publishing exact stats for revenue/traffic/etc for Appointment Reminder. The main reason is that there exists a subset of the market which is less sign-up-on-your-website-for-\$79-a-month and more six-months-of-presentations-and-then-we-sign-contracts-including-an-NDA-and-then-cut-you-a-five-figure-check-every-month. It wouldn't help either my readers or myself to say "Appointment Reminder is puttering around at \$1,200 a month of sales, boo hoo hoo" when the actual truth was "Oh yeah and I've got a \$200k contract from a hospital system but shhh don't tell anybody because the NDA would make me lose my firstborn if that gets out."

Do you think Appointment Reminder will be as successful as Bingo Card Creator? What are your current plans for it? Have you learned anything new about building a micro-isv you can share, vs. bingo card creator?

I would be disappointed if AR had BCC levels of success in 4 years. I'm pretty happy with where I am in

life right now, don't get me wrong, but I've *done* run-a-scrappy-little-shoestring-business. Time for new challenges. New challenges might not be "Build the next Facebook" but it is equally not "Build the next Bingo Card Creator."

What has been your biggest lesson so far with Appointment Reminder?

It's amazing how much easier it is to consistently work 5 hours a week when you have 5 hours of free time than when you have all the free time you could ever want.

Are Bingo Card Creator and Appointment Reminder your full time projects now? What else do you?

I do about ten to twelve weeks of consulting every year, largely on SEO, marketing, and conversion optimization for software companies. I never really planned on doing this, but people who I respect enormously started offering amounts of money that I just couldn't say no to to do projects which were crazily interesting. I probably won't be doing that for forever, but it's great fun for the moment, lets me travel to the US semi-regularly, and helps to fund AR development. (e.g. I can pay decent American-scale wages to the guy doing my web design right now, and eventually I'll probably bring on folks to handle the programming so that I can focus on marketing/sales/etc.)

What advice would you give to a developer that is just about to begin a micro-isv side project?

Ship! Ship! SHIP! The single common thing among all entrepreneurs you admire is that they got over their mostly irrational fears and obsessions and actually shipped something to customers in return for money. I've talked to literally hundreds of people who aspire to being micro-ISVs (i.e. to run small software businesses), and far and away the biggest failure mode is that they never actually release software for money. (#2, with a bullet, is releasing something but not marketing it.)

SHIP!

HOURS Tracker



iPhone

Idea: Create a time tracking app for iPhone.

Project Maker(s): Carlos Ribas

Download App: <http://itunes.apple.com/no/app/hourstracker-time-sheet-time/id321923934>

Tell us a little about how you got started in web development and software.

I've been a web developer since the early days of the web. Before that I was doing Windows development. Over the years I have worked on various development projects ranging from e-commerce sites, to finance web applications, to high-performance transactional systems.



What were you doing before you decided to build a side project, and how did Hours Tracker come about?

I was working as a .NET Software Architect at a small company responsible for banking products. The motivation for my side project was simply that the App Store boom was a major feature in the news at the time. There were stories of people writing things like iShoot and being able to quit their day jobs and work for themselves. That sounded really appealing, so I decided to give it a shot.

Why did you decide to launch this idea specifically?

My wife actually had a real world need for a decent time tracking app. She had recently purchased an iPhone and her search for a time tracking app turned up mediocre solutions. I thought a time tracker would be a good basic app to help me get familiar with iPhone OS, so I started work on it.

Once you decided to do this, how did you literally get started?

I purchased a Mac Mini, signed up for the developer program, and downloaded maybe 10 sample apps from the Apple developer website. I played around with those and then started planning the time tracker.

If you started it while you were working. How did you find the time to do both, was it difficult?

I mostly worked on my side project during the weekends and sometimes weeknights. I would say it was somewhat difficult just because you need to stay motivated and be willing to give up your leisure time. iOS development is great in that the costs are low. You can get a Mac Mini or used Mac for under \$600 and upgrade to better hardware as you get more serious.

What is your business model?

My business model is simply product-focused. HoursTracker solves a small set of problems but aims to solve them simply and proficiently. So many apps try to be all things to all people. I don't believe that appeals to the average smartphone user. They want simple and easy. My strategy with regards to product design was to follow Apple's lead. They provide great resources such as the Human Interface Guidelines, sample apps, and of course, the built-in iPhone apps.

When you launched, how long was it until you saw your first sale?

HoursTracker sold 34 copies the first day it was in the store, just from organic/search discovery in the App Store. I certainly did not expect to sell more than maybe 3-5 copies, so when I saw those first figures I started taking the app a lot more seriously. App Store search still remains the primary driver of sales. I don't spend much money on traditional marketing because I have never seen results in any test I've ever done.

Can you walk us through the tools you used to build the initial version?

HoursTracker started out with XCode 3 and iOS 3.0 beta. It is pure objective-C and uses hand-coded SQLite for data storage. The initial version took about 4 months from start-to-finish, working very limited hours.

What parts of development do you do and what parts do you hire out?

I perform all aspects of development.

What has been your biggest surprise so far?

The biggest surprise is the overall success of the app, and also the fact that new apps that are obvious copycats seem to fail. That makes me happy. I didn't copy anyone's app when I created HoursTracker and I expect the same from other developers.

What has been your biggest mistake?

The free version of the app got to a point where it had too many features and was discouraging users from upgrading. When the ad revenues plummeted, this became a problem and the free app had to be removed from the store altogether.

What tracking tools do you use? How do you use them?

I use Google Analytics on the web and Flurry Analytics within the app.

How is your project doing now, and what are your plans for the future?

HoursTracker is still far from feature-complete. The original plans cover a larger feature set and use-cases than I've been able to build into the app. I plan to get the app closer to feature complete and then see what other opportunities present themselves :)

What advice would you give to a developer that wants to work on their own side project?

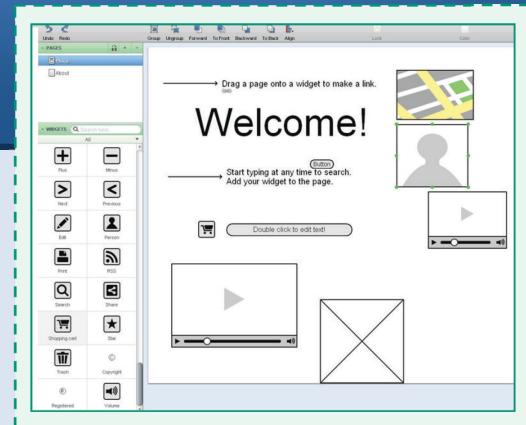
Go for it! Software development is somewhat unique in that the overhead cost can be almost entirely labor. The ones-and-zeroes don't actually exist -- you can use as many as you want and even tear it down and start over. It's not like wood working or mechanical hobbies. So, if you have an original idea and development skills you should definitely build a side project and enjoy the adventure of selling directly to customers via the massive market that is the app store.

Mockingbird



Browser

Idea: Create an easy way to build and exchange wireframes online.
Project Maker(s): Saikat Chakrabarti
Website: <http://www.gomockingbird.com/>



Tell us a little about how you got started in web development.

I made my first website in fourth grade during the summer of 1995 after picking up a book at the local used bookstore. I believe it was the promise of definite fame and fortune that made me buy the book -- one of my friends had made a website about DeLoreans by this point and had promptly become a superstar in the online community of DeLorean enthusiasts (all twenty of them or so).

The book had some absurdly scammy sounding title like "Make totally cool websites with HTML!" (I can't seem to find it on Amazon, nor do I remember the actual title), but my nine-year-old self tackled that book like it was Shakespeare's greatest work, reading it and rereading it to grasp whatever deep meaning it must have had. I have a very distinct memory of being extremely elated when, after about a week of struggling to turn a blank box with a red x in it into an image of Pete Sampras, I finally succeeded. I think that joy -- seeing something happen as a result of something I just typed -- was what first got me hooked on programming.

That first website eventually expanded as my friend with the DeLorean website and I merged and we made our ultimate startup -- a website where we told the world our erudite thoughts on tennis players, DeLoreans, and lightsabers. It was going to take over the world.

Between then and my first real job out of college, I mostly dabbled in programming -- making websites for high school or college organizations or competing in programming contests. It wasn't until I started programming with others -- some in college, but mostly in my first job -- that I felt like I started to actually understand how to build real software.

What were you doing before you built Mockingbird?

The first job, referred to above, followed by 6 months of attempting to make other startups that never got anywhere. The job was as a technology associate at a hedge fund called Bridgewater, and I was living in New York. It was the first place where I learned how to architect complex software systems and actually design things. I quit about 10 months into the job to try to start my own business, and during this time Sheena (my Mockingbird co-founder) and I went through about three or four other ideas. I eventually moved to San Francisco, mostly on a whim (it seemed like the place startups were happening). At the time, Sheena was still in New York and we soon realized how annoying it was to exchange wireframes for the idea we were working on at the time. Thus was born Mockingbird.

How would you recommend someone who has a full-time job that wants to build a side project get started?

I actually started Mockingbird a few months after quitting. I did, however, try to work on a startup on the side while I was at Bridgewater but didn't find the time to do both (my job at Bridgewater was about 12 hours a day including the commute). We had very few initial costs (just a Linode box for twenty dollars a month), and I still think the best thing I did while I had a full-time job was to save as much money as

possible. Since I didn't really succeed in having a full-time job and a side project at the same time, my advice would probably be to just quit and try to pursue your side project full time with all your might, but this only really works once you have some savings and if you have the freedom to take a financial risk like that. If you don't and have to work full-time, don't have a twelve hour a day job.

We launched Mockingbird as a free beta about five and a half months after working on it. The launch was pretty much accidental (I asked some of the people in the Cappuccino community for feedback and they tweeted about the site and basically launched it for us -- I spent my first week after that sleeping by my laptop and waking up every hour to check to make sure that Mockingbird was still up), and traffic on the first few days was pretty high. We had about 80,000 unique visitors in our first week (about half of those on the first day), which eventually settled to a steady five thousand to six thousand a day.

The Mockingbird we launched our beta with looks a lot like the Mockingbird of today (minus a lot of features), but the very first prototype I made was way uglier.

What was your big differentiator? Why did you decide to launch this?

When we decided to make Mockingbird, there were no other web applications that made it very easy to both build and share multipage mockups easily. Our initial goal was simply to make something where we could share mockups with a link, and link together multiple pages of mockups with nothing but drag and drop. The ease with which we allowed people to share and link together mockups was our biggest differentiator when we launched and this ease of sharing and collaborating is what we still focus on today.

What is your business model?

We charge monthly for the service, and it has turned out to work pretty well. We haven't tested a bunch of different strategies yet though, so there is probably room for improvement here.

So when you launched, how long was it until you saw your first sale?

After we launched the beta, we stayed as a free beta for about a year and then launched our paid version of the product. As a result of this, by this point we had a lot of beta users (around sixty thousand) so we made our first sale pretty quickly after launching (I think it was in about 10 minutes). We did not spend any money on marketing at all, and all the marketing we got was word of mouth and through Twitter. Our initial costs were just money for Linode, Pingdom and then a few fees to setup our LLC (including everything, less than a hundred dollars a month initially). This turns out to be not very far from the costs we still have today.

Can you walk us through the tools you used to build the initial version of Mockingbird?

We built the very first version of Mockingbird in Cappuccino with Django on the backend. It was served using an Nginx reverse proxy on a Linode 512 box. We used PostgreSQL for our database. To make a prototype of Mockingbird took about two weeks, but it took about five and a half months to make the version we launched with (and I'm still not sure if it was too long or too short or just the right time).

How did you deal with technical issues?

I googled a lot and asked a lot of questions on IRC. I didn't have very many technical friends when I started Mockingbird (and only knew about three people in San Francisco when I moved here), so IRC ended up being the main place I looked to get help from people who actually knew things.

What parts of development do you do and what parts do you hire out (costs?).

Sheena and I do all the development on Mockingbird.

How are you getting users for the product?

We continue to mostly get users through word of mouth. I can't share too many specifics, but we're still growing at a pretty good rate.

How was your growth during the first three days of launching?

The first three days after our beta launch and after our paid launch were both three of the biggest days of new users for us. I don't have specifics unfortunately, but I believe we got about three to four times our

normal number of users on the first few days after our paid launch.

What has been your biggest lesson so far?

There have been quite a few. It's hard to pick one, so I'll list a few and be annoyingly pithy in the process:

- * Start charging for your product sooner rather than later. I'm still not sure if doing a completely open beta was in the end better or worse for us, but I do think we did it for too long.
- * Focus on one or two features and do those well. In fact, focusing on a few things at a time as a general rule served us well, especially in light of our constrained bandwidth
- * Try to separate your work space from your non-work space

What advice would you give to a developer that is just about to begin a SaaS side project?

I would tell her to pick the one or two things that she plans to make the focus of her application and just build those out first and make them great. Then build out the minimal number of peripheral features that are necessary to make the application not totally useless. Then launch it.

What tracking tools do you use? How do you use them?

We use Google Analytics for tracking page views, Mixpanel for tracking client-side events, and Pingdom for tracking uptime (amongst other notification tools to tell us if Mockingbird goes down).

How is your project doing now, and what are your plans for the future?

We're still going strong and for the near future, just have a bunch of features planned for Mockingbird itself. I still think one of the main advantages we have over others is our simplicity and the very low cognitive overhead to using Mockingbird, and so this is something we pay a lot of attention to before adding anything at all. We intend to keep Mockingbird simple and easy and not really try to expand it to try to do too much.

Simple Physics, Dummy Defense



iPhone

Idea: Create an indie, physics-based, iPhone game.

Project Maker(s): Andrew Garrison

Download App: <http://itunes.apple.com/us/app/simplephysics/id408233979>

<http://itunes.apple.com/us/app/dummy-defense/id460832678>



Tell us a little about how you got started in web development and software.

I've been writing games since I was in high school, around age 15. I started with C and read tons of books by Andre Lamothe. I wrote a few simple games back in high school, my best game was named Scorn, which was about 5k lines of complete spaghetti code. The game was actually quite fun though. It was a side view shoot-em-up game with rocket launchers, grenade launchers, teleporters and more. I think the best way to describe it is with a picture:

Early Game, "Scorn"



It was obvious to me at this time in high school that writing games was something that I wanted to do for the rest of my life.

What were you doing before you decided to build a side project, and how did it come about?

It was early 2009, I had been working at a large defense company, working on a flight simulator. It was actually a very good job, and I picked up quite a few skills with OpenGL and C++. I had also been dabbling with Flash games for a while, but I was frustrated with the low revenue I was seeing. I also dabbled in Silverlight and I even wrote the first version of BridgeBasher in Silverlight. I had very few hits though and I was frustrated with the lack of traction.

However, I kept reading about games like Trism and iShoot, where indie game developers were having tremendous success on the iPhone. I remember one night; I just decided that I was going to get started. I literally said to myself "I'm going to do it. I have to do this now." I drove to 3 different Best Buy stores, searching for a copy of OSX and I rigged my Dell XPS system as a Hackintosh and started writing Skyword. In retrospect, I should have just bought a Mac. Hackintoshes are a time pit and you'll spend more time kludging it to work with the newest versions of OSX than you will spend actually writing games.

If you started it while doing something else, how did you find the time to do both?

I started writing my first iPhone game just before having our first child. Once my son was born, it was much more difficult to find time, and in fact I took quite a break from iPhone development because work and caring for a child just took too much and I didn't have the drive to keep it going. The only initial costs I had were the cost of OSX, which was about \$100. I was using a hackintosh, so the monetary expense was low, but the time investment was more than it should have been in order to keep the hackintosh updated to work with the new patches.

What is your business model?

Business model? I did not have a business model in mind when I started. I just wanted to create an app that at least one other person would enjoy. I have started learning more about the business of app development over the years, and I have learned some tricks along the way. I have found the most effective way to market an app is to make it free. FreeAppADay has been a great help in this area. Joe Bayen, the CEO, directly contacted me and asked if I wanted to make BridgeBasher free for a short while. I was completely ignorant of app marketing at this time, so I was hesitant to do it, but I am so glad that I did. Making BridgeBasher free for about one week yielded about one million downloads. When I switched it back to paid, I made more money in a few weeks than I had made in the past year. This also established a large user base that I was able to tap into when launching Simple Physics.

When you launched, how long was it until you saw your first sale, and where did it come from?

Simple Physics launched on March 29, 2011. I had accrued an email list of about 35,000 people thanks to an email signup form in BridgeBasher. On release, I simply emailed my BridgeBasher players and told them that I had released a new game. I also pushed out an update to BridgeBasher that had a link in it to my Simple Physics. This worked great. I shot up the charts to #7 top overall in about a day. I remember the feeling of seeing my app pass up incredible titles from EA, and other top notch game companies. It was definitely a highlight of my life. The sales figures for being in the top 10 was also quite a treat. Being in the top 10 back in 2011 was about \$10k per day. I was only in the top 10 for a few days, but it was still a huge boon to my confidence and my bank account.

Can you walk us through the tools, languages, etc. you used to build the initial version?

I have written all of my games, except Skyword, in C++. I only use Objective C when I have to, which is when I'm interacting with the Cocoa framework. I actually spend almost all my time in Windows 7, writing code in Visual Studio, and then when I'm ready to release an app, I reboot into OSX and work on getting the software to run on an actual iOS device. This is my least favorite part, though I'm getting quite good at it.

The initial version of BridgeBasher took some time to build. I worked on it in the evenings and weekends for about 6 months before my first version. I then updated BridgeBasher quite a few times and kept working on my game framework and started working on Simple Physics. Simple Physics took me about 6 months to write the first version, which consisted of only 5 levels.

What parts of development do you do and what parts do you hire out?

I do all of the programming and marketing, but I have started contracting out artwork for my newer games. I have an artist that I consistently work with now, and our relationship is easy at this point. He has a good concept of the games I'm building and he produces very high quality work.

How are you marketing it?

Here are the figures from Simple Physics launch, which was back in early 2011. It peaked at #7 on the top overall charts.

Day	Downloads	Chart Ranking
Day 1	180	-
Day 2	4,392	27
Day 3	14,737	8
Day 4	16,145	7
Day 5	14,562	7
Day 6	14,567	7
Day 7	13,510	8
Day 8	9,288	8
Day 9	7,273	11
Day 10	4,988	16

What has been your biggest surprise so far?

The fact that Simple Physics did so well on launch really surprised me. I was extremely excited to the point that I could not concentrate on anything else. I ended up taking the day off and just staring at the app store on my iPhone. Refresh... Refresh... Refresh.

What has been your biggest mistake?

My biggest mistake so far is Touch Island. Touch Island is the third app I wrote. It was a complete flop. Technically speaking, it was probably the most challenging app I've written to date, but the sales have been abysmal. I almost called it quits after such a devastating blow, but luckily BridgeBasher, which had been out for over a year and also had quite depressing sales, suddenly started gaining momentum and my passion was re-ignited. I learned a valuable lesson from Touch Island though, which is to never to develop an app blindly. I had no idea if anybody would buy Touch Island because I never tested the market to see if anybody wanted it.

What tracking tools do you use?

I use AppFigures for tracking app sales. It's a great web app, and I highly recommend it. I need to start tracking user interaction from within my app, which is something that I have not done yet.

How is your project doing now, and what are your plans for the future?

I just released a new app, Dummy Defense, which is doing very well in the app store. It peaked at the #36 top overall app in the US app store. I plan on releasing some new levels for Dummy Defense and then porting some of my apps over to Android. After that I will probably start working on Simple Physics 2, which will include some great features that have been suggested by my players.

What advice would you give to a developer that wants to work on their own side project?

Staying motivated is the hardest part, and it takes a lot of determination to keep going. I didn't write this quote, but it helped me quite a bit to keep going:

"Henceforth, I will consider each day's effort as but one blow of my blade against a mighty oak. The first blow may cause not a tremor in the wood, nor the second, nor the third. Each blow, of itself, may be trifling and seem of no consequence. Yet from childish swipes the oak will eventually tumble. So it will be with my efforts of today."

Once you overcome the drudgery of getting your first app out there, I think it is easier to keep going because it is so exciting to see people using your software. At this point, you must establish a solid user base. The best way I've found to do so is to make your app free for a while, using a platform like FreeAppADay. I was able to get around 100k downloads per day with BridgeBasher while it was free and being promoted by FreeAppADay. Here are the download figures for BridgeBasher while it was free back in 2011, just a couple of months before launching Simple Physics:

Day	Downloads
1 (January 30, 2011)	141,029
2	184,094
3	183,206
4	162,376
5	157,667
6	161,376
7	112,115

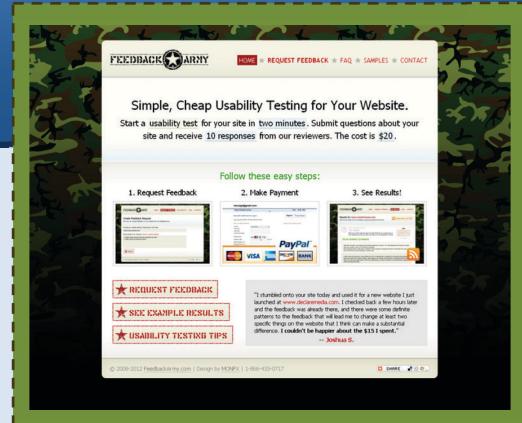
I am convinced this is what gave me my start.

Feedback Army



Browser

Idea: Create an app to utilize Amazon's Mechanical Turk to collect feedback from many people quickly and cheaply
Project Maker(s): Raphael Mudge
Website: <http://www.feedbackarmy.com/>



Tell us a little about how you got started in web development and software.

I've never considered myself a web developer. I started out as a kid on internet relay chat interested in Linux. In the late 90s, I learned Perl while working at an ISP. They put those skills by making me write CGI scripts for web development clients we had. Sadly, my design skills still carry an air of "1997" to them.

What were you doing before you built Feedbackarmy?

Before I built Feedback Army I was an active duty officer in the US Air Force. I read too many Paul Graham essays and decided I wanted to try my hand at entrepreneurship. Feedback Army was a distraction from my "main project" After the Deadline, a software service grammar checker that was picked up by Automattic in 2009.

If you started Feedbackarmy while you were working, how did you find the time to do both?

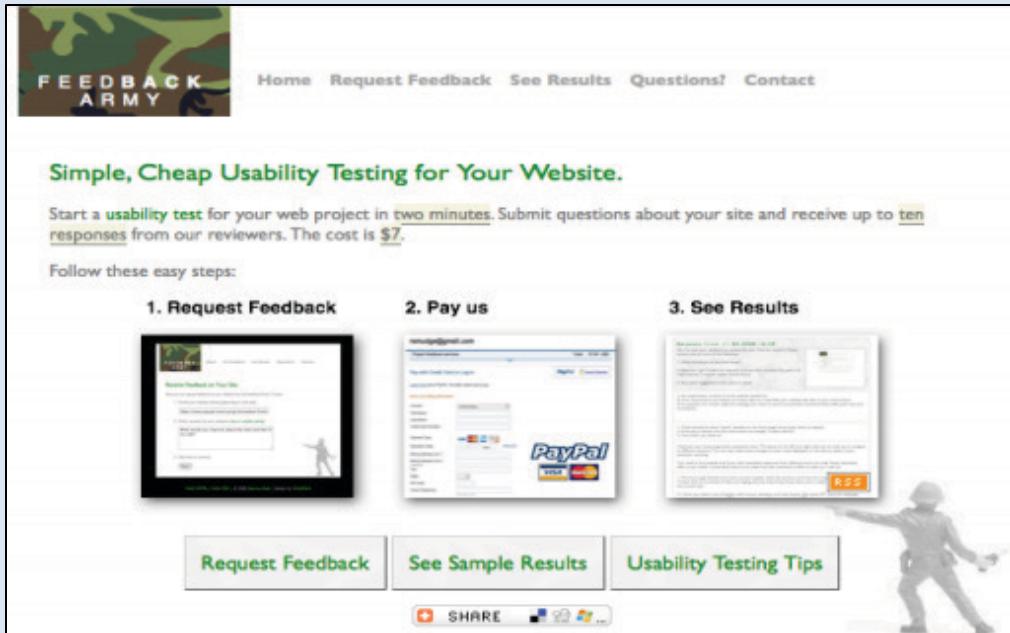
I developed Feedback Army while I was finishing a graduate degree and working on After the Deadline. I started developing Feedback Army towards the end of my last semester and I really wanted something to do other than study. The idea for Feedback Army found me and so I jumped on it. Time wasn't a problem because, as a student, I had plenty of it. It helped a lot that Feedback Army was very simple and well scoped. My other project, After the Deadline, was not.

How did the idea for Feedback Army find you?

I was browsing Hacker News one night and saw a post asking if anyone knew of a website feedback service. Someone posted in response that they used Amazon's Mechanical Turk to collect feedback from many people quickly and cheaply. A lot of people replied to this person asking him how he did it. Reading this post, I saw an opportunity for something I could try to implement very quickly.

What was your big differentiator? Why did you decide to launch this?

Feedback Army's differentiator, at the time of launch, was its price. People were shocked that they could have ten people review their site for USD \$.70 per person. I raised my prices since 2008, but they're probably still too low. I decided to launch it because this was my cyber lemonade stand. Once I found the idea, it was easy to implement, and I didn't really have much to lose. If no one liked it, it would go into the rubbish bin with other projects I tried that didn't gain traction.



Early Version, Feedback Army

But that didn't happen, and it has performed pretty well for a weekend project. How would you recommend someone who has a full-time job that wants to build a side project get started? I keep a journal and I set goals in that journal. Don't think I'm going to tell you "set goals and write them down", because I'm not ☺ For years I set a goal of creating something, anything, and selling it online. In my journal I referred to it as a cyber-lemonade stand. I didn't care if it was big or small; I just wanted to sell something. While I had the worldly comforts and distractions of full-time employment, I never found the time or motivation. The way I truly motivated myself to pursue entrepreneurship was to tell everyone that I knew "I'm starting a business", amass some savings, and quit my job. By tossing myself into a situation where I had no choice but to pursue projects, I suddenly had both time and motivation to do it.

What is your business model?

Feedback Army's business model is very simple. Small business owners, e-book authors, usability practitioners, designers, and many others want to know what others think about their website. I connect my customers with reviewers who look at their site and answer my customer's questions. I pay the reviewers a portion of what my customers give me and keep the rest as profit. Since Feedback Army has always been profitable, I think the model works great.

So when you launched, how long was it until you saw your first sale, and where did it come from? I launched Feedback Army on Hacker News late November 2008. The site did very well on Hacker News that day. I credit this to being active in the Hacker News community and having friends on the site helped the story get an initial up vote boost when I launched it. I understand that the Hacker News community is now a stronger force than Slashdot.org in some cases.

I immediately had ten or so unique customers within the first few hours of launching the site. Early on, the site was setup to send a message to my cellphone when I got a sale. When the launch seemed to be going well, I met a friend of mine for tea and I was quickly chattering on about how things were going. It was a lot of fun because as we talked, my phone kept buzzing with an SMS message indicating another sale.

Did you expect this to happen?

I had no idea what to expect as my other ideas didn't get traction.

In hindsight, Feedback Army benefited from timing. Other businesses, like usertesting.com, came online in this area of cheap self-serve usability testing around the same time. Journalists like to write about markets being disrupted or changed in some way, multiple businesses make it easier for them to prove their point and tell a story. Traditional usability testing, while it has its place, is prohibitively expensive for most projects and this was a market waiting for disruption. I didn't plan it this way, but Feedback Army benefited greatly from this story that journalists and bloggers were ready to tell.

What were your initial costs?

Other than time, launching Feedback Army did not cost me anything. I used a virtual private server that a friend had provisioned. I did the design work and development myself. The initial Feedback Army design was so ugly and amateur looking that I'm still amazed I received press or customers. I would not have bought from me. Fortunately, a talented designer Monjurul Dolon (monfx.com) approached me, thanked me for the positive experience he had as a customer, and marketed his services to make Feedback Army look better.

The real answer is launching the site cost me nothing financially, but my initial design execution was so poor it probably cost me a lot at the same time.

What are your main methods for marketing the site?

I used to purchase banner ads on different sites that I thought would fit my customer demographics. I later came to the conclusion that this form of advertising is nothing more than a nice way to make a donation to the owner of a blog. Sometimes this worked very well as a blogger whose site I purchased an ad on would notice Feedback Army and choose to write a blog post about the service. This method did not work well for attracting visitors and converting them to sales though.

Early into the life of the site I had a lot of success with direct marketing over email and twitter. On Twitter I would search for people asking for feedback and send them a quick note making them aware of Feedback Army. One of my testimonials, zazzie.com, came from a Twitter outreach. I used to also contact potential customers with personalized emails and a coupon that let them try the site for free.

I tried to scale and automate both of these direct marketing approaches with poor results. I wrote a program to harvest feedback requests from Twitter, out source writing a personal response on Mechanical Turk, check the quality of the response with Mechanical Turk, and send messages that pass to the user over Twitter. The results were pretty good but the conversion rate didn't match the cost, so I discontinued this practice after a few days.

For six months, I had a marketing assistant whose purpose was to generate leads based on topics I'd provide. For example, at one point I had my marketing assistant find people who wrote e-books about weddings. After validating the leads, I'd task my marketing assistant to write personalized emails to each of them with a "try Feedback Army free" coupon. This worked and it generated business, but it wasn't enough new business to cover the cost of the contractor.

All of these little stories are a way of saying I never found one marketing method that paid for itself and didn't require a lot of work on my part.

When it was active, my blog was my most powerful marketing tool. For about five months I wrote very candid blog posts about what I learned running Feedback Army, customer stories, and my experiences promoting Feedback Army. Blog posts that hit well with the Hacker News crowd usually generated a spike in sales and sometimes a mention in a prominent blog as someone else became aware of the existence of my business.

I now rely word of mouth to generate sales. Blogs, newsletters, and even speakers at conferences continue to talk about Feedback Army and recommend it to people.

Can you walk us through the tools you used to build the initial version of Feedback Army?

Feedback Army, as with most of my web applications, is built in a perl-like scripting language for Java

called Sleep. I wrote Sleep as an embeddable language for an IRC client many years ago. The language works with third-party Java libraries quite well and corrects many things I perceived as warts in Perl. The application is hosted in an app server I wrote for Sleep called Moconti. I get strange looks when telling people that I wrote my own web stack from the ground up, but I'm very productive with it and it's handled everything I've thrown at it so far. After the Deadline has processed millions of documents each month for nearly two years from the same server using this stack. I didn't know this when I built Feedback Army, but this homegrown stack is pretty robust and works for me.

How long did it take to build your initial version, MVP?

I built the initial version of Feedback Army in about four hours. It took two weeks from developing this prototype to get to the point where I could launch Feedback Army. I had to integrate PayPal, write an FAQ, and experiment with different reviewer pay levels to quickly turn around quality reviews.

How did you deal with technical issues?

The technical challenges surrounding Feedback Army involve quality assurance for the responses. Now I largely put this task on my customers by giving them the ability to reject any response for 48 hours and receive a new one in its place.

However, a big problem I faced early on, was duplicate responses. Some people would answer a feedback request, modify it slightly, and then post it from another account. Some of these modifications were more than slight. It was also hard for a human to spot so customer feedback on this issue would come long after the reviewers had been paid.

To solve this problem I took a nuclear approach and implemented a correlation algorithm to find and score like responses. This algorithm made it very easy for me to investigate responses that were too similar, reject them, and ban the affiliated ids. Unfortunately, this also required a lot of manual review effort on my part.

Now, I have a system that very effectively prevents duplicate responses. It's very simple, doesn't require work on my part, and I haven't had a duplicate response report since implementing it a year and a half ago.

What parts of development do you do and what parts do you hire out?

I do all of the development and I now hire out any design work. I will never ever design any website ever again. If my sister wants a blog, I'll find a designer on odesk.com before I create an atrocity with HTML again.

How are you getting users for the product?

Word of mouth. Sometimes a newsletter or blog will mention Feedback Army and I'll end up with a slew of new customers.

How is growth for the product?

Summer time and December are the slowest months for Feedback Army, but January through April is when I see a period of growth each year. The second year the customer base doubled around January and I still don't know what the cause was.

How was working with mechanical turk api? How was the quality of the results?

I think the Mechanical Turk API and SDK were designed by a ten-year-old genius at Amazon who has never designed an API before. The API isn't well documented, exposes too much complex detail, and does not provide a simple way to execute the most common use cases. Support for the API is non-existent too. Of course this was three years ago and maybe it has evolved. I once had a clock skew issue on my virtual server and all of my calls to Mechanical Turk started failing. I had no idea what the cause was and feared that I would lose my business to this glitch. Fortunately, I got lucky and tracked it down, but it was not a happy day for me. On the bright side, my code continues to work with their service three years later, someone cares about backwards compatibility.

While Amazon does not provide developers with technical support for Mechanical Turk, their soft support is great. When I launched the project, a manager from Amazon immediately reached out to me to make herself available if I needed her help. When she left, her replacement made sure to establish a similar relationship. This is something I really appreciated.

Working with the Mechanical Turk community is fun too. It really is a community. They have their own message board and some people work on Mechanical Turk as a hobby. I sometimes have pushback from people claiming that everyone on Mechanical Turk is from random places outside the United States. While Mechanical Turk is a diverse crowd, you'd be surprised by some of the people who do work through the service. I once gave a talk on the Feedback Army at a library in Arlington, VA. Someone asked me about the reviewers and who they were. A woman stood up in the audience, introduced herself as a freelance writer, and talked about her experiences writing reviews for Feedback Army. She was so well spoken and enthusiastic that some people thought she was a plant. Mechanical Turk is a big community and turkers are everywhere.

As for response quality, I think it's pretty high and believe the service is actually underpriced. Some reviews come back with a terse sentence per question. Others come back with paragraphs of information. In aggregate, the quality of the product is high.

How was your growth during the first three days of launching?

The first days of Feedback Army were strong because people were writing about it and linking to it like crazy. It was on the front-page of Delicious the week it launched. I can't really give you a number that contrasts the first days and the first three weeks though. I'd say I had splash in the beginning and a little smoldering over the first three weeks. The rest of Feedback Army's growth is a product of quickly responding to my current customers, many marketing experiments, and word of mouth.

What has been your biggest lesson so far?

Feedback Army was the first thing I created that made money. This taught me something very valuable. It taught me that money does not always come in a package that ties an hour of my time directly to a number. It's a great lesson and one that is best realized by experiencing it. If you're on the fence about entrepreneurship, I recommend getting something simple going that earns a profit without your presence. It's an addictive experience and will motivate you and develop your entrepreneurial confidence better than anything else ever will.

What advice would you give to a developer that is just about to begin a SaaS side project?

Scope your project to something very simple that you can launch quickly. I also recommend using some beer money to hire a designer on odesk.com. You can find very talented people willing to work for a very affordable rate.

What tracking tools do you use?

I use Google Analytics and I love it. I have the goals mechanism set up so I can track conversion rates and break them down by different sources. This is one of the mechanisms I used to evaluate my return on investment for different marketing expenditures. I highly recommend it.

I also have a spreadsheet that I use to trend sales and life-time customer value. The process is kind of manual, but I go through PayPal each month and account for my sales and log these in the spreadsheet. I also have a mechanism on Feedback Army to track how many times each customer has used the service. This allows me to calculate the life-time value per customer. If my customer acquisition costs are ever less than my marketing costs, then I've just found my scalable marketing solution.

How is your project doing now, and what are your plans for the future?

Feedback Army is doing well in that it continues to serve customers, generate a profit, and receive unsolicited testimonials. I've had a few offers from folks who wanted to buy Feedback Army and even went a few steps with one prospective buyer. This process forced me to think about what I want from the site and I came to the conclusion that I'm not interested in selling the service. I get a lot out of it for what little I've had to put into it. My customers like the service, my reviewers like the service, and I'm quite

happy with that.

My future plans are to continue operating the service, provide great customer service, and implement new features here and there to better serve my customers and reviewers

Gratitude Journal



iPhone

Idea: Create a gratitude journal app for the iPhone.

Project Maker(s): Carla White

Download App: <http://itunes.apple.com/app/gratitude-journal-original!/id299604556>

Tell us a little about how you got started in web development and software.

I studied information systems during my MBA program. After getting my MBA, I worked for a software company as a technical consultant. I managed software development and expanded the practice to Europe, giving me both business and project management experience. From there, I launched my own business in London. That's when I started designing. I later became certified in Human Factors and designed for NASA. Creating iPhone apps started as a hobby, but grew into a full time business over a couple of years.



What were you doing before you decided to build a side project, and how did it come about?

I was working for NASA designing computer interfaces. It was interesting, but a little dry. When the iPhone came out, it was the most premium user interface I ever seen. So I decided to learn how to design for that interface and created my first app. It was only for learning purposes, but proved to be very popular so I created more. Then others started asking me to help them, so I wrote a book and started a business.

Why did you build Gratitude Journal?

My first mobile app was Gratitude Journal. I decided to create it because I had been keeping a gratitude journal for a few months and it really helped me get over the death of my father. I started the business so I could let others know about the power of gratitude.

Once you decided to do this, how did you get started?

I sketched out a few ideas on paper. I talked to a friend who created a small app and got his advice. I also emailed others who were creating apps. I studied Photoshop to learn how I can design the user interface. I came up with the name Happy Tapper while I was out for a walk and launched a website after the app was on iTunes.

If you started it while you were working. How did you find the time to do both, was it difficult?

I worked on my app from 5-7am before going into my day job. I also put in some hours after work and on weekends. It was exhausting and I often thought of giving up, but usually something would happen that told me to keep going. My initial costs were the purchase of a Mac laptop since I only had PC's at the time. I also had to pay for development which was only \$500.

If you have the passion, drive and believe in your idea, starting a business with a full time job is possible. In fact, it's best to have a full time job so you're not too concerned about costs and pulling in a profit straight away. It usually takes a couple years before a business is sustainable, and it's much better to have some steady income while you're going through the teething pains.

What is your business model?

Revenues come from apps sales, book sales, and consulting. I donate 10% of my profits to charity.

When you launched, how long was it until you saw your first sale?

With Happy Tapper it was about 2 months before I received any revenues from my app sales. When I wrote my book, sales happened instantly upon launch. The only marketing I've done is for my books. I've done banner ads, affiliate programs, and of course use social media. The affiliate programs are the best. I also am very generous with giving away product which helps sales.

Can you walk us through the tools you used to build the initial version?

I created the interface of my first app using Photoshop. I also used Photoshop to create my website. I outsourced the development and used Xcode to test any code they provided to me. The first app took me about 5 months to complete, from concept to launch.

What parts of development do you do and what parts do you hire out?

I do all the designs, marketing, and project management. I outsource the development.

What has been your biggest surprise so far?

I'm surprised at how popular my apps are and who is using them. I love that my business has connected with me so many wonderful people around the world whom I would have never had met otherwise.

What has been your biggest mistake?

I trusted some developers way too much and was afraid to fire them when their work was poor. This has resulted in bad code releases and very unhappy customers. As soon as I know something isn't working out, I stop it and change things to make them better.

What tracking tools do you use?

I use Google Analytics a bit but mostly rely on sales information from Apple.

How is your project doing now, and what are your plans for the future?

I'm busier than ever with so many app requests – both my own apps and consulting for other businesses. I continue to focus on life improving apps and am getting more into health care and education.

What advice would you give to a developer that wants to work on their own side project?

Do your homework first. It costs nothing and will greatly reduce your risks.

Wedding Lens



Idea: Create a service for engaged couples to collect high resolution photos from their guests.

Project Maker(s): Justin Chen

Website: <http://www.weddinglens.com/>

Browser

A screenshot of the The Wedding Lens website. The page features a header with the logo and navigation links like 'Why The Wedding Lens', 'Pricing', 'Resources', and 'Sign In'. Below the header is a main section with the heading 'Easy, elegant wedding photo sharing' and a sub-section about the service being easy to use. There are four buttons: 'Collect More Photos', 'Save Time', 'Save Money', and 'Save the Environment'. At the bottom, there's a section titled 'Features' with bullet points and a 'In the Press' section with a quote.

Tell us a little about how you got started in web development and software.

At the beginning of our entrepreneurial journey in January of 2006, my business partner and I had minimal web development experience. We both studied computer science at U.C. Berkeley, but our corporate jobs at Microsoft and HP did not take us down the web route. Consequently, when we decided to quit our jobs and start Menuism.com, there was a considerable learning curve. Not only were we starting our first business, but we were also learning a new programming language and framework at the same time.

When it came time to build The Wedding Lens as a side project in 2007, the technical and product development experience we gained from building Menuism.com gave us the confidence to decide to build another business and the skills to do so much more quickly than the first time around.

What were you doing before you decided to build a side project, and how did it come about?

We were both working full-time on growing Menuism.com, our first entrepreneurial endeavor, when I got married in mid-2007 and asked my business partner for a way to gather all our guests' photos. I did not want to get barraged with links to different online photo services that displayed scaled down photos; I wanted to download all the full-resolution photos to my own computer.

My co-founder used the few days while I was away on my honeymoon to quickly prototype a service that let our friends upload to a single place on the web and download each others' photos. Based on the positive feedback from friends and family we decided to explore building it into a business.

Why did you decide to launch this? What was your big differentiator?

We decided to pursue The Wedding Lens as a business because it was a solution that solved a particular need for customers that were willing and able to pay for the service -- engaged couples. The Wedding Lens is different from other online photo services because its focus is on gathering all the photos around a particular event -- a wedding, and the service's model is not based on printing profits.

The goal of The Wedding Lens is to be the easiest way for the engaged couple to get all the full resolution photos from their guests. This involves making it easy for guests to upload (no registration required, authenticated Evite-style link, bulk uploading, etc), providing notices to remind guests to contribute to the collective photo collection and bulk downloading.

If you started it while you were working, how did you find the time to do both, was it difficult?

Once our initial market research validated that there was a viable market, we decided to take a month-long break from our main business and build out the first version of the group photo sharing service. When we first started Menuism.com, we also decided to quit our full-time jobs to give it 100% as opposed to the work-on-the-side model. Since we run our businesses from different geographic locations, we personally need to have the full-time commitment to motivate us to get things done in a timely manner.

Prior to quitting our full time jobs, we tried to work on things on the side but it was difficult to make progress.

As with any software or web project, the initial costs were pretty minimal. We were already hosting Menuism.com on EngineYard so we decided to host The Wedding Lens there too for an initial outlay of a few hundred dollars a month.

What is your business model?

The Wedding Lens is a SaaS (software-as-a-service) app where the bride and groom pay for six, twelve or twenty-four months of service in lump sums. It was initially a freemium model, but we found that people who sign up for a free product rarely use it properly, don't experience the intended value and end up having a sub-par experience. While we were in the freemium phase, we saw that people would sign up for free and usually upgrade within two weeks, which is typically enough time to play with the service and determine that it suits their needs.

Upon switching to from the freemium model to all paid plans, we took this into consideration and built in a two week free trial to all plans. Rather than a typical monthly subscription, we decided on the fixed time period plans because we wanted couples to have the service long enough after the wedding that they would have an adequate amount of time to gather all the photos (since people are slow at uploading photos) and to enjoy their online photo gallery. As soon as we switched from the freemium model we saw an nice uptick in overall sales and it has been earning a relatively consistent few thousand US dollars in revenue for the past two years.

When you launched, how long was it until you saw your first sale, and where did it come from?

The first sale came about six months after the launch from a friend of a friend. Initial promotional efforts revolved around tapping into personal networks and giving free premium albums to any acquaintance, friend or friend of friend getting married. A number of weddings happened before the initial sale and the viral nature of weddings finally kicked in.

Some of our other marketing initiatives included:

- Reaching out to wedding bloggers and offering free premium album coupon codes which they could use as prizes for reader contests.
- Setting up an affiliate program so other wedding blogs and coupon sites could profit off promoting the service.
- Applying the SEO lessons from Menuism.com.

The initial blog articles we were able to get led to a moderate influx of paid customers and from there we were able to leverage positive word of mouth and SEO to continue growing sales. We tried paid ads on Facebook and AdWords but had a difficult time getting a positive ROI.

Can you walk us through the tools you used to build the initial version?

Since we had used Ruby on Rails to build Menuism.com, we also decided to use it to build The Wedding Lens. In fact, the initial MVP that my co-founder built was a simple hidden page hosted on Menuism.com that allowed friends and family to upload zip files, view other people's zip files and download them. Once we decided to build it into its own service, it took about a month to get the full service online.

What parts of development do you do and what parts do you hire out?

All of the development was done in-house and we continue to handle the on-going technical maintenance. We have one contract designer that we regularly use for screen mockups, HTML and CSS and we have another contractor that handles customer service and social media activities.

What tracking tools do you use?

We primarily use Google Analytics for traffic trending and we are big fans of the event tracking functionality; we use it to do A/B testing and to track on-page events to see how users are interacting with different features.

How is your project doing now, and what are your plans for the future?

The project continues to do well and it takes minimal ongoing maintenance effort outside of the customer support related issues that are mostly outsourced. There are plans to update the service to take advantage of some of the technology advances that have developed in the past couple years. Outside of that, we're always looking for marketing partnerships that can help increase exposure to the wedding market.

What advice would you give to a developer that wants to work on their own side project?

Market focus. As a developer, it's easy to come up with an idea and think that the elegant and scalable solution can solve a variety of different problems. While that is probably true, to run a business an entrepreneur needs to be able to find customers and communicate value quickly and effectively. Focusing the product and efforts on a particular market or use case make this task much easier. When there are the constraints of a single market or customer type, it lets the entrepreneur really dive deep and understand the customer and industry. In our particular scenario, we've learned an incredible amount about the wedding industry and we would have not been able to do that if we positioned the product as a general purpose group photo sharing app.

Be persistent. Coding a product is easy to accomplish in the short term (i.e. startup weekend), but building a business is a long term game. Word of mouth takes time. The requisite brand repetition for recall takes time. Getting to really know the business, industry and flush out a sustainable model takes time. The launch of a product to the world is always fun and exhilarating, but that buzz dies down pretty fast and if there is not a sustained effort to keep pushing the business, it will just die too.

Droid Secret tips



Android

Idea: Create a quick user manual app for Android users.

Project Maker(s): Tim Mackenzie

Download App:

<https://play.google.com/store/apps/details?id=com.simplifynowsoftware.droidsecrettips>



Tell us a little about how you got started in web development.

My day job was as a software developer, so some of the skills for Android development (like Java) were part of my work experience. There were many other skills that I had not honed as part of my past work experience, though, which had to be developed further when I began building apps.

What were you doing before you decided to build a side project, and how did it come about?

I was working on another side project, my first idea for an Android app involving fitness. It was growing in scope, and the time horizon to finish it kept slipping out.

In the midst of that, I had an idea to create a quick app based on information I already had on hand. From that, Droid Secret Tips was born. It was partly a play to get an app released quickly to see how things went, and partly an experiment to see how ads would work for me.

Why did you decide to launch this?

When I realized that I had the material (and skills, from working on my original app idea) to build a quick app and release it, I decided that it would be better to get something out than have lots of ideas on the drawing board.

Droid Secret Tips is simple – it displays information about Android to the user, in a simple to navigate format. I collected the tips when I got my first Android phone, and I have continued to grow the list of information. Now I can make use of that information a second time, by sharing it with other Android users. The advantage to the users is that they don't need to search for the information themselves.

I still haven't launched my original fitness app – I keep adding to what I want it to do, and getting sidetracked by other projects.

Once you decided to do this, how did you get started?

I sketched out a few layouts and decided on the simplest workflow I could come up with. While the app has evolved greatly since then, the first release was extremely simple.

With the work I had started for my other project, tutorials found on the web, and the instructions for the ad SDKs I pulled the app together in about the equivalent of three days. I spent the next few months tweaking it and improving it, but the initial app was functional and available to the world after three days of development.

Part of the ability to get the initial app done so quickly was to minimize the features in version 1.0. Feedback from that release encouraged me to keep going.

If you started it while you were working. How did you find the time to do both, was it difficult?

I already had an Android phone and a computer capable of handling the development environment, so

there were no costs there.

The one cost you can't get around is entry into the Android Market (now Google Play), but that's only \$25. While you can just release to the alternate markets, you are missing out on a big chunk of Android users.

Other than that, while I spent money getting a web site and filing a legal business entity, that won't be necessary for everyone.

Time, on the other hand, was a matter of determination. I enjoy writing apps, so I squeezed a few hours of development in whenever I could. Progress is progress, and eventually things come together.

Working a full-time job does limit the number of hours that you have available to write apps, but if you have sufficient patience and determination to keep going, remember that you're in it for the long haul.

What is your business model?

I am a big believer in multiple streams of income. Not only does it protect you in case you have problems with one source, but it also helps even out when one increases or decreases.

Revenue from most of my sources varies widely from month to month. I do paid app sales as well as advertisements. The levels of revenue from each ad network keep changing, which causes me to adjust where I send my traffic.

App sales also vary widely, affected by things such as app releases, holidays, and many difficult-to-measure factors.

It took a few months to reliably make over \$1 per day for my first app, but I suspect that higher-use apps like games will see higher revenue per install because each user spends more time in the app (and viewing ads). The selection of ad types also can greatly affect the revenue per user.

When you launched, how long was it until you saw your first sale, and where did it come from?

I did no marketing for my first app release, but I immediately saw downloads of the free version. Two weeks later, I launched an ad-free pro version, which saw 14 downloads the first day! I attribute that to two things: appearing on the 'new' list, and from users of the free app (who now finally had an upgrade path).

I don't remember where those sales were from, but I find that 60-80% of my sales come from the USA. Apps with a more localized appeal may do better in other locations.

Since then, I have introduced other forms of marketing, including social media and press releases. For the types of apps I have created, however, I haven't been able to track enough results from these campaigns to determine that I should spend more time or money on them. I generally spend more time making my apps polished and have a great user experience, which should lead to positive ratings.

Can you walk us through the tools you used to build the initial version?

As I mentioned, I built my MVP in about 3 days of effort, and I was happy to ship it as quickly as possible. The immediate traffic was gratifying and kept me motivated.

I used Eclipse with the ADT, the vanilla development environment recommended by the Android team.

The free app started with a few hundred downloads a day, which seemed great to me. Over the next year, that download count grew to over 70,000.

What parts of development do you do and what parts do you hire out?

I do all development and just about everything else, except for some of the graphics. This has kept my expenses low, at the cost of my time. This means it is harder to get as much done on apps, as there is a lot of non-development labor involved in getting an app out the door.

I would certainly suggest hiring out to get a good icon, as apps are judged by their icon. Your feature graphic should also look good. Stock art may be a good and inexpensive option, but be sure you read the licensing terms.

How are you marketing it?

I promote my apps through my website with a description page with app store badges. I also promote my apps through twitter, and have worked on getting reviews. It is very difficult to get a response from review sites, by the way.

My niche holiday apps haven't done nearly as well on Google Play as on some of the other app markets I use, but they still gain momentum with a little push. I do some advertising through house ads, which connects users of my other apps to the new one. I have done some paid ads, but am not satisfied that I get a good return on my money from them.

My latest push is for free ad-supported apps on Google Play, and paid apps on the other app stores. It's important to understand the buying and app downloading habits of each market, because the customers can be very different.

It's also important to note how easy it is to get lost in Google Play – there are hundreds of thousands of other developers. Marketing is not optional.

What has been your biggest surprise so far?

When Amazon decided to promote my app, I found out just how powerful their marketing engine is. They cut the price in half and promoted my app through several channels, and sales skyrocketed. In the Amazon Appstore, momentum is key - being in the top 100 increases sales, which keeps you in the top 100. For a 12-hour period, Droid Secret Tips Pro was the #1 selling app on the Amazon Appstore.

What has been your biggest mistake?

I accidentally released a version of Droid Secret Tips Pro on the Amazon Appstore that had a major bug in it. By the time I got the fixed version through their QA process, I got two scathing one-star reviews. My app never regained the same popularity after that.

What tracking tools do you use?

There are so many tools to use, and none of them do everything for me (at least none of the free offerings).

My favorite tool is still the first one I used – homegrown A/B testing for ads, where I randomly select which network to use. This helped me to initially determine which network provided the best return for ad traffic.

Unfortunately, it gets much more complicated when you have a lot of ad units and apps. I keep investigating sites like MoPub and tweaking my tracking strategy with the best tools for what I'm currently working on.

Tracking takes effort, as does deciding what you want to track and measure.

How is your project doing now, and what are your plans for the future?

Droid Secret Tips is trucking along nicely, and does fine with minimal maintenance. I do have plans for a major overhaul, but I have several other projects in the works to complete before then.

I continue to work on my holiday apps, because they are fun and sell decently well on the other Android app markets.

Aside from development, a lot of my focus is in teaching now. In particular, I spent a lot of time sharing my experiences into my blog at <http://ProjectJourneymen.com>, to help other developers benefit from my experiences.

This has grown into the Android Income Series of books (<http://AndroidIncomeSeries.com/>), which has books on topics from starting your app to selecting the right ad networks and app markets for your apps.

What advice would you give to a developer that wants to work on their own side project?

The best advice I have is starting small and get it out the door quickly. Don't shoot for the moon and never release. Seeing downloads and getting feedback is a great motivator to keep going and make the next step.

Textual App

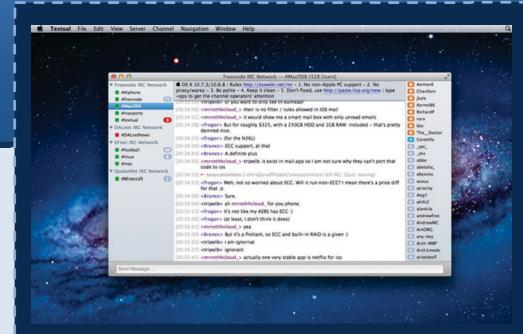


Mac App

Idea: Create an app for my wife to automate the process of adding up multiple order items with taxes and shipping.

Project Maker(s): Jeff Weisbein

Download App: <http://itunes.apple.com/us/app/textual-irc-client/id403012667?mt=12>



Tell us a little about how you got started in web development and software.

I started my first website (<http://www.besttechie.net>) at the age of 13, which I still maintain to this day, after convincing my parents to let me use their credit card to buy the domain. Everything I've done since then (including Textual IRC) has kind of been under the BestTechie umbrella. Most of what I know is self-taught or learned from reading and a lot of trial and error.

What were you doing before you decided to build a side project, and how did it come about?

Being that I started fairly young, I was in school while working on my projects. BestTechie came about because I realized in 2003 (when I started the site) that the web was going to be amazing platform for sharing information and knowledge. The Textual IRC project came to be because I was a longtime Windows user who loved mIRC and I could not find an acceptable alternative for OS X.

Why did you decide to launch this?

After about or a year or two of running mIRC in a virtual machine and testing just about every native Mac IRC client, I decided something better had to be created. I wanted something that didn't use a lot of system resources, had good theme support, had awesome scripting support, and had a great community of users behind it.

Once you decided to do this, how did you get started?

I'm not too much of a programmer, so I reached out to a good friend of mine to help me make Textual a reality. We started talking about features and UI and decided it would be best to take an existing IRC client that was open source, make our improvements and go from there. That's exactly what we did and it's worked out very well so far.

What is your business model?

We sell Textual in the Mac App Store for \$4.99. Originally, we offered it as freeware/donationware, however, when the Mac App Store came to be we saw it as a good opportunity and something to take advantage of (being that it's installed on all new Mac's). So we made sure we were in the store on opening day with our latest version (at the time) 2.0. This was also a major upgrade so it worked out well and our existing user base was more than happy to pay for it. Using the Mac App Store has worked out very well to date, in fact, we often place within the top 10 paid social networking apps.

When you launched, how long was it until you saw your first sale?

Our first sale was essentially immediately upon launch of the Mac App Store. We rely on word of mouth marketing and also include a link to download the application in the default quit message for Textual.

What has been your biggest surprise so far?

The biggest surprise for me thus far has been the number of sales we're doing. I did not expect an IRC

client to be so popular, but am definitely thankful it is. I'm also very excited that people love the product enough to recommend it to their friends. I've learned that both Leo Laporte of TWiT and Joshua Topolsky of The Verge use Textual.

How is your project doing now, and what are your plans for the future?

Textual is doing well and we are currently working on an update to last major release. The update fixes a number of bugs and adds a number of features requested by users. We plan to stay focused and continue to develop and improve the product.

What advice would you give to developers that want to work on their own side project?

My advice would be: don't spread yourself too thin. This is especially true if you have a full-time job. Pick one side project and do it, if it isn't working out and you have another idea you want to work on dump the one that isn't working and focus on the new one.

It's also important to have some free time in your life and not always be working. I learned this the hard way after being burned out. When you're constantly working or thinking you lose the ability to come up with fresh new ideas or new perspectives on things. By setting aside time for yourself to relax (watch TV or a movie, read a book (not a technical one), listen to music, go outside, exercise), you will end up being more productive and probably come up with more (and better) ideas. I have also found that going for a walk outside can be useful in clearing your mind and is great for relaxation.

DNSimple



Browser

Idea: Build an app that allows you full control over your domains.

Project Maker(s): Anthony Eden

Website: <http://dnsimple.com/>

The screenshot shows the DNSimple dashboard with the title 'Add DNS Records for aetnion.com'. It lists several services with their respective record counts and preview links. Each service has an 'ADD' button next to it, with some already marked as 'ADDED'. The services listed are Google Apps (23 Records), Heroku (4 Records), Posterous (2 Records), and Blogger (4 Records).

Service	Records	Description	Status
Google Apps	23	Add Google Apps for your domain. Includes Docs, Spreadsheets, Gmail, Calendar, Presentation and more.	ADDED
Heroku	4	Use Heroku as your web host.	ADDED
Posterous	2	Use Posterous for your blog.	ADD
Blogger	4	Use Blogger for your blog.	ADD

What were you doing when you started Dnsimple on the side?

I started development of DNSimple in April of 2010. At the time I was working with several other developers in Hawaii doing freelance web application development.

Is Dnsimple your full time project now?

No, currently DNSimple is only a side project. I hope to change that when it reaches a point that it can sustain my family and I.

How did you come up with the idea?

I've been working with domains and DNS since 1999. The first time was at one of the first domain registrars to be accredited by ICANN when the .com top-level domain was deregulated. After spending several years working at that registrar I went on to build the domain registry for the .mp top-level domain. I operated the .mp registry and registrar for several years before taking a break from the domain industry to work in government. In 2009 I returned to work on the .mp top-level domain again and that brings us to the inception of DNSimple. In 2010 I decided I wanted to build something that I was truly responsible for, where I would be making the business decisions in the way I felt was best for the business. I was using GoDaddy and was getting really annoyed by all of the up selling, and I noticed a lot of my friends felt the same way. I looked around at the various domain and DNS providers and realized that, aside from a few notable exceptions, most registrars were providing a pretty awful experience.

Had you built any other side projects before this?

Yes, I've built various side projects. In fact just before launching DNSimple I built and launched PillowTalkr, a game for couples to improve communication (<http://pillowtalkr.com/>). Unlike previous projects, however, DNSimple is the first project that I've launched with a real focus on becoming a profitable company.

Weren't there already domain name services like this out there already?

Absolutely, there is a ton of competition. The domain industry is already 12 years old (perhaps even older if you consider hosted DNS, although prior to '99 there were relatively few hosted DNS providers). Since it is a mature industry by Internet standards I knew there was no way I'd be able to compete on price so I decided to compete on the experience. By experience I mean both the experience of registering and managing domains and DNS and the customer support experience. I knew that an API would be an essential part of the application and so I have spent considerable time on that. I also knew that simplifying domain registration down to as few steps as possible, with no up selling, would be a key means for making customers smile, so that was part of what went into DNSimple as well from the very beginning.

How did you get started after your first thoughts on the project?

I've been developing software for a long time and I have a bunch of tools that I can use whenever I need to develop something, so I just jumped right into it. I started a spike in Ruby on Rails and just worked

through the basic models and UI. Once I had the basics working I paid a designer to come up with the basic site design, color scheme and logo and then put it all together.

Did you work with a partner, or was it just you?

I work with my brother, who is my business partner on DNSimple. He handles the system operations while I handle most of the coding. Occasionally he writes code for the app and occasionally I deal with operational issues, but usually we each handle our own tasks. For the design I have worked with two different designers, one who handled the initial design and one who helped improve some elements of the user experience. The second designer was the one who came up with the layout and flow for the services screens, which people really like. I continue to work with both designers off and on as time and money permits.

What tools did you use to build it?

I develop on OS X and deploy to Linux (Ubuntu). The application runs on Rails 3 with a PostgreSQL database as the primary data store. I also use Redis thanks largely to Resque, which is an excellent background job queueing system. The DNS runs on PowerDNS with MySQL backends that are slaved to a master MySQL database.

What things did you have to learn for this project?

I had to learn several new APIs, although fortunately I already understood the business model so it wasn't exceptionally difficult.

How did you deal with technical issues?

With rigor and patience. :-)

How long did it take to get your first version up and running, or MVP?

We launched 3 months after inception.

After building, how quickly until you saw your first customer or sale?

The day we launched we had customers, mainly because I talked to the community that I am most comfortable with and kept them up-to-date with what I was working on.

Do you know where it came from?

Our first customers were friends who knew what I was working on. From there new customers started coming in from my Twitter followers and their followers.

How did you start or continue marketing it after that?

Directly after the launch I was posting to Twitter on a regular basis and speaking at conferences. Launching the API was a good way to market the service because I was able to go out to Ruby conferences and actually *show* people a cool command-line tool for registering domain names. I also submitted to several startup-specific blogs and then received a significant boost in traffic with a post on Hacker News.

How was your growth during the first three weeks of launching?

It was slow. We were getting a couple new signups each day with no marketing aside from posting to Twitter but only 1 in 6 would actually subscribe.

What about the first 3 months?

The growth was fairly steady, but still small. Every day we would get a couple people trying out the service but not everyone would subscribe. DNSimple didn't actually offer domain registration and transfer until the end of September in 2010 so initially the customers were limited to those who really wanted hosted DNS.

During the first 3 months we had a little over 100 people try the service and out of that about 1 in 4 actually subscribed.

How is your app doing now and what are your future plans?

DNSSimple's customer count continues to grow, which I am very pleased about. In July of this year we reached a major milestone of 1000 paying customers. There are lots of things to improve about the domain transfer experience, so that's a big item on my list. Additionally I continue to think about ways to add additional top-level domains without sacrificing the simplicity of the DNSSimple interface. My main goal is to continue providing a service and experience that makes my customers happy.

Do you consider yourself a designer or developer?

I am a developer.

What are some of your other side projects? How did you come up with those ideas?

As I mentioned earlier, PillowTalkr is another project that I worked on before DNSSimple. That idea was a joint effort between me and my wife. I was explaining Gowalla to her and she pointed out that there were no apps that were targeted towards couples. We brainstormed a bit and came up with PillowTalkr and I built it in a month.

Do you have any guidelines or rules for coming up with ideas for side projects?

Whenever possible, solve a problem you actually have or that someone close to you has. You are your best customer and worst critic so ideally you build something you can use. I think it's possible to build something for others if you know a market well, however most developers do not understand markets outside of their development space. For example, there are huge opportunities to improve things like HR, time tracking and expense reporting, but very few developers actually know and love those markets, so you get products that are designed for features, not to solve problems.

What advice would you give to someone wanting to build their own iPhone app side project?

I'm not really qualified to give that advice, but my feeling right now is that you shouldn't be developing a project for a single device; rather you should be developing for the web. It is possible to do some pretty amazing things with HTML and JavaScript. It's essential to have a good API that can be used by devices, but I still believe that the web is a good platform to start on.

Ventrilode



iPhone

Idea: Create a voice chat application that lets users connect to their favorite ventrilo servers via iPhone.

Project Maker(s): Charles De Meulenaer

Download App: <http://itunes.apple.com/us/app/ventrilode/id486115720>

Tell us a little about yourself, and how you got started in web development and software.

My name is Charles De Meulenaer. I'm the creator of Ventrilode (\$4.99), a third party Ventrilo client for the iPhone that has been downloaded several tens of thousands of times. Ventrilo is a hugely popular Voice over IP system used by millions of gamers all around the world.



I initially got into development when I was 12, when I spent a few days on building a simple website with Front page to administer my bookmarks. Well, if you can really call that development!

Once I got the hang of the Front page drag-and-drop interface I decided to take a look under the hood, at this mythical language called HTML. I played around with it for about a year, trying out different things and building stuff that just seemed cool. At a certain point I got in touch with some people who were building dynamic websites and this seemed really interesting – so many potential use cases! They pointed me towards some PHP/MySQL 'getting started' tutorials and off I went! The next two years of my life were dedicated to learning both programming in general and PHP. The great thing was that I had more than enough time to learn programming properly, since I was still in high school at this point. I've found PHP to be a simple language which allows you to easily get the hang of most programming concepts.

Programming for the web was interesting, but after a few years I wanted a change. The next years were spent learning different programming languages, such as Java, C/C++ and Python. Eventually I was hired as a C/C++ game (technology) developer. I spent 1½ years doing that, before venturing out and becoming an entrepreneur.

In terms of using past skills; for Ventrilode I used my background of C/C++ to write the audio related code. For the rest of the app, I used the normal iPhone development environment (Objective-C and Cocoa).

What were you doing before you decided to build a side project, and how did it come about?

I previously developed a pretty successful app for Android and had just spent a few weeks doing consulting work for a client. Once done with this, I decided to take a look at some of the ideas that I had written down over the previous months. A Ventrilo client for the iPhone seemed to be the one with the highest chance of success and thus I decided to build it!

The idea initially came to me after I had purchased my first iPhone. I was spending a lot of time hanging out with friends on Ventrilo and noticed that sadly, there was no application for it. My previous phone was an Android phone that did actually have an app, so imagine the disappointment when I didn't have this anymore! I also had played World of Warcraft a few years back and remembered that there'd be frequent posts on various forums asking for a Ventrilo app.

Why did you decide to launch this?

I had several ideas for different apps, but the main reason why I chose this one was because I had some initial market validation. Several friends with new iPhones had asked me whether there was an app like this and I had to disappoint them!

Combining the amount of forum posts where people were asking for this app that I had seen over the years, the amount of people that were asking me directly as well as the total amount of people worldwide using Ventrilo (several million), this seemed to be enough to go ahead and build it.

The biggest advantage was that the focus was on building a niche application. It was something that people really wanted - the app is something that solves this exact need.

The million dollar apps (\$0.99 per unit) are the ones that cater to a wide audience, but that's also the area of the app store with the most competition. There are business apps with tens of thousands of sales at \$15-30 price points. People often tend to forget, the ultimate number of download your app gets usually doesn't matter if it has no effect on your bottom line (except for additional exposure) profit.

Once you decided to do this, how did you get started?

The biggest issue with this application was doing the integration with Ventrilo. After about a month of research, I managed to get a simple two-way voice chat going and after this it about a month was spent on the design and implementation of the UI of the application. Once released, another month was spent fixing bugs.

If you started it while you were working. How did you find the time to do both, was it difficult?

I'd just finished doing a consulting contract and wanted to develop another app. I've found this worked great for me, as it allowed me extended periods of freedom to develop new apps/technology. It also allowed me to self-fund most of the development work. I've recently written in a blog post about the difficulty of combining a FT job with a commercial side product and I think contracting/consulting is a great way to go if you're serious about building a successful app.

I didn't have any initial costs since I was also doing mobile consulting. However, I did have a MacBook pro, an iPhone and an iOS developer account. If you don't have those, then you'd probably be looking at an investment of about \$2500-\$3000.

What is your business model?

Ventrilode is a paid application, one time purchase of \$4.99. This really made the most sense for this type of application. Making the app free with the possibility of upgrading after using it for a while could be an option as well in the future, though.

In terms of revenue numbers, currently not disclosing those. However, the app has been downloaded several tens of thousands of times. Initially we had the release week where the app was priced at \$2.99, but those don't really account for an excessive amount of downloads.

When you launched, how long was it until you saw your first sale, and where did it come from?

Hah, this is interesting. I remember the app being approved and then entering the app store at 10pm (GMT +1). Since it was getting pretty late, I just decided to get a good night sleep before doing a marketing push. When I woke up, I had two emails in my mailbox thanking me for releasing this app. The sales for that (short) day were 16 units.

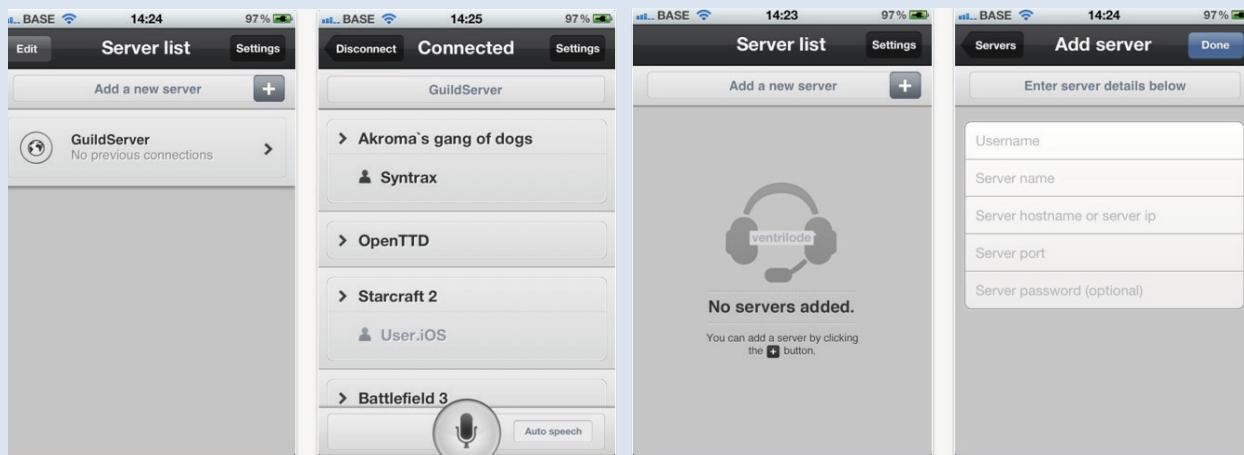
Can you walk us through the tools you used to build the initial version?

Nothing special really! For the development of the app I used Xcode 4. The UI of the app was built with interface builder for the most part, with some custom drawn cells using Quartz2D for all the channels. The audio driver was written entirely in C and the Ventrilo interface in C++.

The entire v1 took about 2 months, with another month or so to fix major bugs and add/change recommendations from users (several settings).

What parts of development do you do and what parts do you hire out?

I did all the actual development work. The design of the app was done by a great UI designer (who is also a good friend of mine) for which he gets a certain percentage of the total app revenues.



Ventrilode UI Design, Artwork

How are you marketing it?

Revenue is currently not being disclosed, but I can tell you that we've sold several tens of thousands of units. This includes a week at \$2.99 and the rest of the sales being \$4.99.

In terms of marketing, we've actually done a few things.

- Make sure we end at the top of the Google search results/in forum threads when people look for terms such as "Ventrilo iPhone", "Ventrilo app", ...
- Lots and lots of posts on forums
- Some in game advertising in World of Warcraft (Yes, we travel through the distant world of Azeroth in search of customers!)

What has been your biggest surprise so far?

Most of the people using the app really love it. Once it was first released (about 2 weeks before Christmas), I got several emails thanking me and saying that Christmas came early. That gives you amazing feeling!

What has been your biggest mistake?

My initial thought was that there'd be a lot of coverage from different blogs. But alas, breaking through the waterfall of apps that are being released every day has proven to be hard. However, because people often play with a group of friend's sales have been fairly decent. If one person in a group finds out about it, he/she often tells his/her friends.

However, if there would've been more blog exposure from high profile blogs I believe sales could've been a lot better. So there's that, and getting featured of course.

What tracking tools do you use?

In all fairness, the application is quite simple so there is not really a big need to log things. However, some of the things that we do log are unique installations, amount of connects and the total amount of people using the application on a daily basis. Most of this is handled by a simple system that was built for this.

How is your project doing now, and what are your plans for the future?

Pretty good, sales numbers are at a steady level. I think this is a successful app release.

In terms of what the future holds, I'm not exactly sure yet! I've developed two successful apps now – one for Android and one for the iPhone. It might make sense to pass on some of the things that I've learned doing this through some sort of workshop/coaching to some people (If someone is interested in this – get in touch!). But yeah, there will definitely be more mobile stuff coming in the future.

What advice would you give to a developer that wants to work on their own side project?

Do it, it's an amazing experience and can change your life.

But when you do then don't be delusional. Chances are you'll probably not make a dime, so when you quit your job make sure you can do some contract work/consulting to sustain yourself.

Making money with apps is harder now that the gold rush is over. That said, there's still a huge amount of money to be made in the market but you need to build a polished and feature complete product. The quality of your MVP needs to be really, really high.

AreMySitesUp



Browser

Idea: Create a simple web app to monitor if a web site is down.

Project Maker(s): Richard Felix

Website: <http://www.aremysitesup.com/>

A screenshot of the Are My Sites Up? website. The main page features a central message: "When your websites go down, be the first to know." Below this are sections for "Easy website monitoring" and "How it's work?" There are also promotional sections for "Are My Sites Up White Label" and a "Mobile White Label". On the right side, there's a small image of a smartphone displaying the service's mobile application.

Tell us a little about how you got started in web development and software.

been building things on the web for years, and my work always revolved around computers, so my past/current work definitely provided me with the skills to build Are My Sites Up.

What were you doing before you decided to build a side project, and how did it come about?

I had a full-time job as a Systems Engineer writing custom software at a document management company.

Why did you decide to launch this?

My good friend Chris Coyier (of <http://css-tricks.com>) and I wanted to build a web app together that would help the web community and that was something that we would use. Chris did the design and front end code, and I did the backend/logic. We'd previously made a web app together named Punchcard (it was a time-tracking application) that never really made it off the ground, so we knew we could work well together and had a good workflow; we just needed to make something that had legs. We thought, "Everyone has a website, and it's terrible when they go down and you find out from someone else, so let's see what we can do in that space". Our big differentiator has always been "Easy website monitoring" We don't have a ton of crazy features, just the most popular ones, and this allows us to keep prices low.

Once you decided to do this, how did you get started?

After we figured out what we were going to build, we literally just started knocking out the interface and the site checker code. We just sort of passed code back and forth to each other on our development server and we had a good first version pretty quickly.

If you started it while you were working, how did you find the time to do both, was it difficult?

I didn't really have issues finding time to do both my full-time job and to make AMSU, mostly because I don't really sleep. It's easy to find a lot of time to work on side-projects when you only sleep 4 hours a night and are really into your idea. I know that this is pretty cliche, but working on something that you're passionate about / interested in is a big factor. If you're doing it for the money, you'll probably lose steam. I tackled it mostly because of the technical issues involved in building/scaling a service like this. It was a fun problem to solve, and writing code to solve problems is my version of playing with LEGO blocks. We had pretty low initial costs; we just rented a virtual server and got going.

What is your business model?

Our business model is simple: to make a service worth charging for, and then to charge reasonable prices for it. Unlike most companies in the space, we charge yearly for our service, but people don't really mind because we're established and it's low-priced. We started up with a freemium model, but got away from that for new users because every free user is a drain on resources for a service like this, and we are trying to keep the service stable, reliable and fast for all of our users. We do still have thousands of free

users, but those are all grandfathered in because they've been with us forever.

When you launched, how long was it until you saw your first sale, and where did it come from?

Haha funny story. Our biggest mistake when we launched was not realizing how many users you could put on a server and still be able to have good check times for everyone, as well as how to price, period. We launched with a free plan that gave people something like 60/100 sites for free, which was completely ridiculous. Who wouldn't jump on a service like that? I don't even have 60 domains. That basically crushed the service. Eventually we figured out the sweet spot for pricing, and we haven't really moved from that.

What are your main methods for marketing the site?

Our main method for marketing the site has been word of mouth and blog posts/magazine articles. We have been around providing a good service for a while, and so we are talked about and linked to from quite a few blogs/blog posts. We do have an active affiliate program as well, but most of our sales come from people who read articles about us and try us out or people hearing about us from their friends/co-workers/employees.

Can you walk us through the tools you used to build the initial version?

The front end of AMSU is Ruby on Rails. The back end consists of a bunch of code in a few languages that was written/tested/perfected over a few years' time. AMSU launched and worked great for most sites, but there are always edge cases that you run into once real people start using your service and plugging their sites in. Some sites respond a lot more quickly than others. Some sites respond in a certain way to your requests, and others respond in a completely different, unexpected way, etc. We identify these cases and have written code to handle them as they've come up.

What parts of development do you do and what parts do you hire out?

I do all of the development these days. The new front page was designed by my friend David Link. Eventually the interface will be redone as well.

What tracking tools do you use? How do you use them?

We use GoSquared (<http://gosquared.com>) for stats.

How is your project doing now, and what are your plans for the future?

It's reached cruising altitude (it's stable and successful), so now I'm working on building things that people have asked for, such as an Android app (I released version 1 of that about a month ago).

What advice would you give to a developer that wants to work on their own side project?

Pick something that you'll learn something from even if it is a failure in the marketplace. I can't tell you how many times I've started a new project and have been able to see the dots that connected the experiences from my past projects and how they've gotten me to the project that I'm working on. Keep at it, and release something as soon as you can, so you don't spend too much time building the wrong things.

Photo 365



Browser

Idea: Create an app to let you save a photo a day, for a year.

Project Maker(s): Benjamin Hsu

Download App: <http://itunes.apple.com/us/app/photo-365-remember-your-year/id454629399?mt=8>



Tell us a little about how you got started in web development and software.

I knew nothing about iOS development before I got started. I just knew it would be cool to have an iPhone app. I've had the iPhone since the very beginning but wasn't until 2010 that I wanted to create an iPhone app.

What were you doing before you decided to build a side project?

I work in the family restaurant business. It's not what I want to do forever and that's why I started to think about how I could make enough money so I can quit my job.

I've been trying to make money online in some way for many years. I've tried but quit when I wasn't making much or got bored of it.

I have a passion for all Apple products and thought my own app would be a cool thing to have.

Why did you decide to launch this?

The idea just came to me one day. It had been on my iPhone for months, I just didn't realize it. I had been using an app similar to Photo 365. However the more I used it, the more I realized that it could be done better. While the features were nice, there were no graphics. The calendar page could be improved as well.

I could tell it was done by a knowledgeable person, but spent no money on graphics.

So what I did was took a look at the reviews. I focused on the negative ones. I wanted to read what people wanted. What features they requested. What could make it better? I made notes and that's how I came up with it.

Once you decided to do this, how did you get started?

Well I am not a programmer. I don't know a single line of code. I've tried to learn but it never went very far. So I had this idea but it sat on the side while I figured out how I could get it made.

Then I read a blog post about outsourcing app development. I had never outsourced anything before. I didn't know you could outsource someone to create an app for you. This was my solution! If he did it, I could do it.

So I bought an e-book he recommended that showed me out to outsource the project. That was so helpful taking me step by step from choosing a developer to putting the app in the App store.

If you started it while you were working, how did you find the time to do both, was it difficult?

I just used my free time to work on it. The beauty with outsourcing is that it saves you a lot of time. If I tried to learn how to code the way they knew how, it'd probably take me years. Hiring someone else

saved me so much time.

Yes there is a cost involved. That's the tradeoff. I have much respect for those developers who create their apps with their own hands. If I had the knowledge, I'd spend the time to do it. Plus I would save myself a lot of money.

However in my situation, that wasn't an option. My cost was \$1900 for the graphics and coding. I had the same company do both for me.

I also had to pay \$99 for the developer's fee to Apple. Plus I paid \$10 for a theme for my website. That was it.

When you launched, how long was it until you saw your first sale?

AppAnnie, the site I use for tracking shows I had \$2.80 on the first day, which was a Wednesday. Then on Thursday I did \$79.86. That was so exciting to me!

As for marketing, I did none. I know that's probably what you don't want to hear but I didn't do any marketing. I didn't know what to do, plus I read that paid marketing didn't really work for many.

All I did was have a Twitter account for Photo 365. I had my own personal Twitter and Facebook. I just sent out a message to all my friends that my app was available. They were pretty excited and shared it.

I had a few blogs contact me about doing a review on it. One was AppAdvice.com. Others were smaller ones but I was happy to give them a promo code and do a review. But I think overall it was maybe five total. It was less than ten for sure, I can't remember the exact.

How long did it take to build your initial version, MVP?

From hiring my developers to putting it in the app store took seven months. Now I realize that is a long time. When they gave me the proposal they said it'd be done in 1.5 month. However I had learned to expect it to take longer. I just didn't realize it would take so long. Part of that is my fault because it would take me days to test out a new build and give my feedback on it.

Thankfully they were really patient and didn't rush me. It helped me as a first time developer and having it take so long actually paid off.

What parts of development do you do and what parts do you hire out?

I hired them to do all of it. Since they had a graphics team, I went with them as well as the coding.

What impact has being a featured App by Apple had on your app's success?

It happened to me starting the second week my app went live.

So for the first eight days my app was for sale, I earned \$1201.09. I couldn't believe it. I thought I'd be lucky to make a few sales. My overall goal was to at least make back the money I spent. At this rate, I was going to make back my money really fast!

However it had peaked at \$363 on Saturday and by Wednesday it has slowly dropped to \$63. So it was going in a direction I wasn't happy with. I thought my app would soon fall off the radar.

Then on that Thursday my app got featured on New and Noteworthy. Obviously it was a huge surprise!

Not only was I was featured but I got the #2 spot right above the fold!



So from Thursday - Wednesday I did \$9363.54 in sales! My highest day was \$2294.21. To this day, I still can't believe my luck.

I knew I would only be featured there for a week.

But on Tuesday during that week, I got a strange email from Apple marketing. They asked for art assets for promotion. I read the email so many times because I wanted to make sure I understood what they wanted. I did a Google search, but didn't find much information about it. I knew this was not a common situation.

They couldn't promise they'd use it but wanted it in case. Now, I've learned others who have gotten the email, sent in the art assets and they were never used. So it wasn't a guarantee.

They wanted graphics in Photoshop and in different sizes and they wanted it in less than 24 hours! Thankfully that day was my off day from work. Also thankfully I have a brother who is a Photoshop wizard. If he wasn't, I don't think I could have gotten what they needed.

So my brother helped me come up with what they wanted. I looked in the App store on my iPhone and on my laptop and saw the big ads they show when they feature an app. We went for a similar style.

I emailed the contact person a few times and she was really helpful in answering my questions.

So on Wednesday I sent it in and waited.

Thursday I couldn't wait to see the updates in the App store. I knew it was around 3pm EST when they updated it.

I was at work and it wasn't updated yet. So I went to a forum where I had help from an experienced app developer about sending in art assets. I opened his private message and saw he said "congrats!"

I quickly went to the App store on my iPhone and I saw this.



I had a friend take a screenshot on his computer to see how it looked.



For the first time in my life, I jumped up and screamed as loud as I could. I was at work but we weren't opened yet. Everyone must of thought I was crazy.

I've never won the lottery, but I imagine this is kinda how it feels.

I shared my excitement online and with my family.

Work that night was a blur. I couldn't stand still and I had a huge smile on my face all night long.

I couldn't wait to see the sales the next day. In fact, every single morning it felt like Christmas day. I would get up early just because I couldn't sleep.

From that Thursday till that following Wednesday I earned \$15,493.92. I was making an average of over \$2,000 a day.

Before my app, I've been trying to earn money on the side for years so this made it more sweet.

My first app made it all the way to App of the Week. That's a spot many individual developers don't make. I knew I had luck on my side so I'm thankful for how everything fell into place.

What has been your biggest surprise so far?

So many surprises to me. I'm surprised by how my app has caught on. It's wonderful to get email from users who say they love my app and refer it to friends and family. To me, that word of mouth, is the best because they're enjoying the app so much they want to tell others about it.

Another thing that surprised me is how much money there is to be made in the app store. Not every app is going to make money, but for those that do, they can make a lot of money. Enough to make it their full time job.

What has been your biggest mistake?

Biggest mistake so far would be listening to the users too much. Let me explain.

When the first version came out, people loved it. But I was getting emails and reading reviews about people wanting to save more than one photo a day.

That was tough because the idea was to just save one photo a day. That's what makes a 365 photo project so unique. Of course you'll take more than one photo a day, but you just have to pick one to save.

The requests became overwhelming it seemed and for the next version, I had my developers add multiple photos a day. No more saving one large beautiful photo. Each photo would be a small thumbnail that would be enlarged when you clicked on it.

I thought users would love it, but they hated it.

My mistake was I was listening to those who wanted more than one photo a day, but I completely forgot to think about people who loved one picture a day. Now I started to hear from them. They hated the new style. They were going to stop using it.

I thought I screwed everything up. I failed to keep the original uniqueness of the app.

So for the next version, we allowed users to choose between saving more than one photo a day or saving just one large photo. Now users are happier.

How is your project doing now, and what are your plans for the future?

The app is still bringing in sales every day, but it's far from the big days when I was featured. My app has been downloaded over 55,000 times. Pretty good for my first app.

I've continued to update the app to make it the best Photo 365 app on the market. I have some big ideas for it that I hope to put into place before the end of 2012.

Now I'm finally started to create more apps. I look forward to working hard on making that next big hit.

What advice would you give to a developer that wants to work on their own side project?

I'd say do it. If you've been thinking about wanting to create an app but haven't, you better do it or else you'll regret it later on.

If you're going to outsource your app, I would suggest that you spend money that you can afford to lose. You have to look at it like gambling. You're putting money on the table and hoping for a big return. You might lose it all, break even, win a little, or hit it big.

For me \$1,900 was a lot of money but at the time it was money I would be okay spending. If I didn't make a single penny on my app, I could still pay all my bills. I wouldn't go into debt. Also I would view it as a learning experience.

So spend what you can afford to lose. I don't guarantee my success to others looking to create an app. It's hard to know what's going to be a hit until it gets in front of users.

Photo 365 found a lot of success in a short period of time, but many apps hardly make any money. If you think about the 80/20 rule, 20% of the apps make 80% of the money. I might have even read that it's 10% of the apps, but you get the idea.

But you won't know unless you try.

<http://photo365app.com>

My blog : <http://getbusylivingblog.com>

Twitter: @benny_hsu

RideWithGPS



Browser

Idea: Allow cyclists a way to analyze activity after logging them with a GPS.
Project Maker(s): Cullen King
Website: <http://www.ridewithgps.com>



Tell us a little about how you got started in web development and software.

I've been into computers as early as I can remember. I started really getting into them after installing my first Linux distro in 1995. I didn't get into development until later in life though. My first real development project was Ride With GPS five or so years ago! I liked riding motorcycles and I had a riding buddy, who is actually my current business partner. We wanted a way to map our motorcycle rides, so the project was born.

What were you doing before you built Ride With GPS?

I spent my free time playing with cars and motorcycles. I really started working on Ride With GPS after moving to Corvallis to go to school. Before heading to OSU I spent 3 years at community college. I spent more time getting distracted with work and socializing. Moving to OSU and dropping all my commitments really freed up a lot of time to work!

What does Ride With GPS do?

RWGPS is a business with two sides. We have a direct consumer side, the site you see, and a business to business side which we haven't spent any time showcasing.

On the consumer side, Ride With GPS is a way for cyclists (or anyone with a GPS unit) to analyze their activities after logging them with a GPS unit. The cycling community is a fantastic market. They are willing to spend money on fancy bikes, bike computers with GPS, power meters etc, so \$10 a month to analyze that data and provide motivation is nothing after spending \$4000 on their equipment. Cyclists are also really fun to program for, because the amount of data is great! The log files from modern bike GPS units can contain heartrate, pedal speed (cadence), and instantaneous power output at any point if they have ponied up \$1500 for a power meter. With that data you can infer an incredible amount of data!

Additionally, we offer businesses with a way to embed tracking functionality in their own site. We offer a client side javascript API that interacts with our backend, allowing businesses to embed all the functionality Ride With GPS.com offers into their own site, without the need to create the complicated backend required. We handle all the data processing and storage, the client just needs to put together the front end on their site. It's working out really well for several companies.

Aside from the Javascript API, we also offer a complete turnkey solution, which is a themed version of what you see on Ride With GPS.com, completely run and maintained by us. Actually, as I write this I am wrapping up two busy weeks in Norway! We have been putting the finishing touches on a deal we put together at the beginning of the year,

If you started Ride With GPS while working or in school, how did you find time to do both?

RWGS was started when I was in school. I was working a 20 hour a week student development job in addition to 16 credits of computer science classes a term. I spent probably 10 or 20 hours a week working on the site at first, getting my feet wet and teaching myself ruby and the rails framework. I didn't do much outside of work, school and RWGPS! I did all my partying and socializing before going to OSU, so it was easy to ignore most of the social aspects of school. I don't think RWGPS would have ever got off the ground if I had spent more time with my peers.

For anyone out there with fulltime commitments outside of their startup (school, work, family etc), you have to be prepared to put everything else on hold. You work your day job, sneak time during your day to day activities to brainstorm, plan, sketch out ideas etc, and then all your spare time is spent implementing them. If you think you can have fulltime commitments and build a company in the evening, without sacrificing your social life, you are in for a big surprise!

What was your big differentiator?

We liked motorcycles and planning motorcycle rides, and then we discovered road biking was a great way to get exercise, which had similar mapping opportunities. Our tool turned out more suited to biking than motorcycling. Launching the project to the public was easy – we had a functional tool and the market didn't have much. There were a few other similar projects out there at the time, but we wanted to jump in and share with other people.

What is your business model?

Our business model is direct consumer sales on Ride With GPS, as well as B2B licensing deals on our core software. RWGPS acts as a showcase for what our software can do both the turnkey complete site and the JavaScript API we offer. Our business model has naturally fallen into place. We realized cyclists would be willing to pay a monthly fee, and there is no shortage of cyclists in the world. Our B2B side got started when we received a cold email from Motorcycle USA. They were looking for a mapping solution and didn't want to spend the developer money to inhouse it. One of their developers happened to be a cyclist, so they recommended our mapping site as a possible solution. This was our first deal, and really sparked the business. The owner of Motorcycle USA (motorcycle-superstore.com and motorcycle-usa.com) turned out to be really cool and helped us through the initial process of negotiations, etc. It was a very easy learning experience!

So when you launched, how long was it until you saw your first sale, and where did it come from?
Our first "sale" was actually a donation. I realized we could fund out operation out of some donations, barring development cost. After outgrowing a tiny shared hosting account on a friend's machine, Zack and I ponied up \$2k and put together a nice system. We had it racked, which was only \$50 a month. Donations easily covered that cost, and slowly repaid our hardware investment. In the first year we accepted donations, we received \$11k total. With our small traffic numbers, and the realization that donors are a small percentage of users who are willing to actually pay to receive something, it struck us that we needed to finally release some paid features!**Can you walk us through the tools you used to build the initial version of Ride With GPS?**

As far as tools go, we have a pretty simple development environment. Both Zack and I are VIM users, so linux computers with our software stack plus VIM is all we need. Our app is a Rails app, recently upgraded to Rails3. We use mysql for the boring stuff like user profiles, gear pages, etc. We use Mongo for the geospatial data. Since our geospatial data is mostly handled in JavaScript for presentation, we transfer it around as JSON anyway. It just made sense to use Mongo for storage. We also use Redis for misc stuff like online user counts, simple referrer tracking etc.

The final piece is the hardware stack. Most startups talk about and are advised to do the whole "cloud" thing. After some very simple spreadsheet math, we realized that a cloud based infrastructure was considerably more expensive than owning our own metal. There's a sweet spot your business needs to be in to enjoy the benefits of scalable virtualized servers. We have predictable growth and intense computing requirements, not to mention IO needs, so owning our own machines was a no brainer. If I can give one piece of advice to startups out there figuring out their infrastructure it's this: don't just jump

on the virtualization bandwagon because it's what you are "suppose to do". Determine your growth, your actual needs (compute power, redundancy) and make a decision.

How did you deal with technical issues?

We fix them!

What parts of development do you do and what parts do you hire out?

All development is done in house by Zack or I. When it becomes feasible, we will hire out design work first and then go from there.

How are you getting users and revenue for the product?

Revenue is currently sufficient to support our operations (hardware, bandwidth and our pay) as well as a small ad words campaign. We get most of our users through word of mouth, and our B2B license deals have all been cold calls we have received! Our growth has been very steady – we never had a launch event or any serious press, etc. It's just been a slow upward climb to get at our current 10k uniques a day. We are however getting a upward tick in our growth, and we expect some big numbers to come during next year's bike season.

How was your growth during the early days and weeks of launching?

One user a day, then 3 users a day, then 10.....very slow :)

What has been your biggest lesson so far?

Starting a business kicks your ass and you better be prepared to sacrifice. I have no idea how I managed to maintain my relationship before my girlfriend and I moved in together. I guess she is more understanding than most! Also, I missed out on steady income for two years, to scrape by on donations off a "silly web app". It seems much more worth it now, however the first two years of really trying to make it work rather than getting a job after graduating were pretty lean...

What advice would you give to a developer that is just about to begin a SaaS side project?

It takes five years, no joke and no exceptions. Your business won't even start to stabilize until much longer than you ever expect it.

What tracking tools do you use?

Google docs, and more recently basecamp. Having a simple checklist and milestones is about all Zack and I need to stay on track, but really, we just fly by the seat of our pants. Google analytics is a must of course, and the occasional a/b testing is handled by Patrick McKenzie's a/bingo rails plugin.

How is your project doing now, and what are your plans for the future?

I am writing this from Oslo, after spending two weeks here on a client's dime. So, not too shabby :) Plans for the future are to make this business successful. Maybe another lifestyle business after this one plays out :)

Dream Track Nation, Tiny Plane



iPhone

Idea: Create an indie game for the iPhone.

Project Maker(s): Mathieu Roy

Download App: <http://itunes.apple.com/us/app/tiny-plane/id504600499?mt=8>



Tell us a little about how you got started in web development and software.

Since I'm 14 years old, I've been interested in starting a business online. At first, I was blogging, creating some websites, like all computer-oriented teenagers. When smartphones started appearing, I quickly wanted to get into this business. As soon as the iPhone released, which was one of the first one with serious gaming capabilities, I quickly jumped on the bandwagon and started developing my first game.

What were you doing before you decided to build a side project, and how did it come about?

I was an electrician, working on construction sites.

Why did you decide to launch this?

I always loved games with cars and motorcycles. One of our inspirations was ExciteBike, for the NES. We decided we wanted to make a modern version of it, with a new twist obviously, crazy physics, and it quickly became a project.

Once you decided to do this, how did you literally get started?

I bought books on iOS development, and quickly got the hang of OpenGL, the technology behind most iOS games.

If you started it while you were working, how did you find the time to do both, was it difficult?

My first game took about 6 months to develop, due to time constraints of my day job and family life. It was a hobby, a hobby which would take most of my free time.

When you launched, how long was it until you saw your first sale, and where did it come from?

As soon as we launched our first game, Dream Track Nation, it was featured as iPhone game of the week by Apple. The first day resulted in thousands of sales.

What parts of development do you do and what parts do you hire out (costs?).

I'm the businessman and programmer of the company. The only part we don't do onsite is the sound, which we hire companies known to make some awesome sound effects / music for games.

How did you start marketing it?

We decided to affiliate with Chillingo, a producer / publisher for iOS games.

How has growth changed over time since the launch?

The game is still doing fine, being in the top 100 of racing games on the iPhone. We update it often to attract user's attention, and it seems to stay stable.

What has been your biggest surprise so far?

My biggest surprise was definitely getting iPhone Game of the Week on our first serious title. We were very happy.

What has been your biggest mistake?

Releasing a game that wasn't ready for the public. Games need to be well tested, across all devices. They also need to be well tested against user input / difficulty, as we get too good at our games. My very first game was reviewed to be excessively hard.

How is your project doing now, and what are your plans for the future?

We released, our new game, Tiny Plane, on the App store. We also have more coming soon this summer, and plan to do this as long as we can!

What advice would you give to a developer that wants to work on their own side project?

Be patient, the first release won't necessarily be a success, but from quality game to quality game, you will make a name for yourself, which will upscale your future releases.

Project Bubble



Browser

Idea: Create a tool to manage projects and tasks.

Project Maker(s): Stu Green

Website: <http://www.projectbubble.com/>

A screenshot of the Project Bubble website. The header reads "project bubble" with "HOME TOUR PRICING & SIGNUP". Below it, a testimonial from Stu Green says, "I have not seen another browser-based project management tool that is so clean, elegant and simple to use. It just lets you get things done!" A green button says "See Plans & Pricing" or "read more". To the right, there's a section titled "The easiest way to manage your projects" showing a video thumbnail for "homedesign" and a "Click the play button to see it in action" button. Below this, a statistic says "Over 500 businesses & 18,000 users manage over 60,000 projects with Project Bubble". At the bottom, there are logos for various clients like AMERICAN BIBLE SOCIETY, CONTRACTPAL, WHOLEGRAIN DIGITAL, multiBrain, relish, PXLYUE, and OVERCOMER.

Tell us a little about how you got started in web development and software.

I was a developer/designer with a focus on user interface design and user experience. I had been building websites and web applications for ten years prior to starting Project Bubble. When it came to building my own application I pretty much had most of the skills I needed, however I was about to learn even more.

What were you doing before you decided to build a side project, and how did it come about?

I was doing contract work designing and building websites for various clients in the UK and the US. I needed a tool to manage projects and tasks, but also one that could build invoices automatically off the projects. There were a few out there already but I had specific requirements in the way I wanted it to work, such as I wanted to be able to manage multiple projects on one page and be able to drag and drop them in to order of priority.

I had just heard about CodeIgniter, which is a PHP framework for rapid application development (similar to Ruby on Rails), and I was itching to try it out on a new project. I figured it wouldn't take me that long to build a tool that did exactly what I needed it to do. I knew I would learn a lot in the process and I was eager to learn new skills and put them in to practice, so Project Bubble was born in the summer of 2009.

Why did you decide to launch this?

In the late summer of 2009 I decided to make the project public and see if anyone else would find it useful for their project management, so I opened a public beta and within 3 months had nearly 5000 users thanks to some blogs featuring it. After the public beta I realized it had financial potential and decided to make it in to a business.

If you started it while you were working, how did you find the time to do both, was it difficult?

Bootstrapping your business is very tough. You have to worry about paying bills as normal, but also you have to eat in to your personal time for development. I was working throughout the day doing contract work, and then evenings and weekends would be spent building my application. Because I was learning and it was all very new to me I enjoyed developing it, but after a while it took its toll a little on family life as you would expect. Having a wife who understands what I do has been amazing because she stood by me throughout this initial build period and allowed me to work those extra hours.

Once the business picked up we were gradually able to phase out the contract work and spend more time on Project Bubble. It took a few months, but once this happened it was very exciting because we could block out whole weeks and treat the application effectively as another client project that was paying me for a week or two.

I do remember those early few months were tough, but I knew that in a year or so we'd be reaping the rewards as we are now. I was greatly encouraged by reading blogs and connecting with other boots trappers who were going through the same thing, and this helped me greatly to keep going.

What is your business model?

The business model is SaaS (software as a service) where people subscribe to use the software on a monthly basis. There's no contract and people can upgrade to a paid plan or downgrade to a free plan at any time. The pricing is tiered, so the higher plan allows more usage of the system (e.g. more storage and more projects) which is ideal for bigger teams. There are affordable plans for startups and freelancers who are just starting out.

At this point in time (2 years after launching the beta) we're now able to employ 2 staff, and have a bit left over for marketing, development and hardware which is a dream come true. We're growing at a good rate too.

When you launched, how long was it until you saw your first sale, and where did it come from?

We started charging about 4 months after launching the beta because everyone was comfortable with paying for the service. I initially undercharged because I didn't feel comfortable charging people to use my software, which I realize now was a big mistake. It wasn't for another 9 months or so that I finally decided on the pricing which we have now and have had for the last year. I learnt a lot in that first year about pricing. Once the revenue was flowing we were able to spend on SEO and other methods of marketing such as PPC and social media. Although the marketing budgets were small in those days the rewards were very high and the return on investment was great.

Can you walk us through the tools you used to build the initial version?

It took about 3-4 months to build the MVP. I worked with another designer briefly for the brand, but everything else I built and designed myself. Project Bubble grew massively in features in its first year so by the end of 2010 it was a solid competitor in the market.

What parts of development do you do and what parts do you hire out?

Most of the development is done internally, mostly by me, however sometimes we'll hire developers for the more complicated aspects of the application, e.g. APIs and synchronization with third party software (e.g. Google Apps). As we grow in revenue we'll hopefully do more hiring like this because we've got a lot of exciting development plans. When trying to keep everything in-house you'll save a lot of money and that's been our biggest advantage.

What has been your biggest surprise so far?

My biggest surprise and delight is always when customers say how much the service has changed their lives. To build something that people actually use and enjoy is a great pleasure for any developer and when you read testimonials like the one below it really surprises you that you have been able to contribute to their productivity and happiness.

"I needed a way to manage multiple simultaneous projects, multiple clients, invoicing, major and minor to-do items and working with collaborators.

Project Bubble has been a total godsend. In addition, their unwavering devotion to client care and continuous improvement completely bowled me over. I'm as impressed as I am grateful... and organized!"

What has been your biggest mistake?

My biggest mistake was probably not valuing my product enough to actually set the correct pricing in the early days. Sure, the service was not as fully featured as it is now, but it was still worth a greater value than I placed on it in those early days. I would always encourage other founders to really think carefully about their pricing before setting it in stone.

What tracking tools do you use?

Mostly we use our own, but also we use [Google] Analytics also.

How is your project doing now, and what are your plans for the future?

Project Bubble is growing really well. We have nearly 15,000 users and nearly 500 businesses using the service and our plans are basically to keep offering really great customer service, a super easy-to-use and affordable product for teams and for freelancers.

What advice would you give to a developer that wants to work on their own side project?

I would just say "Go for it!" If you can find the spare time to develop the project, just do it and have fun. You'll learn a lot as you go and you'll gain new skills that you can either apply to your project or to your client's projects. Who knows, you may end up being the next Jason Fried.

My ethos is all about keeping things simple and offering great customer support. Those two foundations are really the key factors in driving our growth and make us good at what we do. We may even apply it to other products in the future.

The Little Crane That Could



iPhone

Idea: Create an indie crane simulation game, for iPhone.

Project Maker(s): Bram Stolk

Download App: <http://itunes.apple.com/us/app/the-little-crane-that-could/id408257810?mt=8>



Tell us a little about how you got started in web development and software.

I've been honing my programming skills, pretty much from the age of 12 (then on my Sinclair ZX Spectrum). I am now 42 years old, so I would say I've had 30 years of practice. During that time, I also got a MSc in Computer Science. So the skills were all there. All that was missing was the courage to quit my job and believe in myself.

What were you doing before you decided to build a side project, and how did it come about?

I was working in what we call a 'AAA' game studio, doing specialized engineering on console titles in large teams. I've always felt that I was very unproductive in such an environment. I was having more fun with spare-time projects making stuff for the iPhone. When I switched to an iOS developer studio, I felt my situation did not improve, and that I could do a better job myself, I had enough and quit my job only weeks before my then girlfriend, now wife, was due with our first baby. She was not thrilled about this, but I felt I had to go solo or 'indie' as we call it in the game development community. The fact that some small titles I made for iOS were already bringing in some money (not enough to live off, but enough to pay rent), made me confident enough to do this. It was strange: in the 28 years I was lacking the courage to quit my job, I finally took the plunge with my first baby due any moment.

Why did you decide to launch this?

Once independent, I decided to do what I was best at. When playing with game-technology, I always enjoyed making physics simulations. I like things like sliders, hinges, force, and torque, mass and such. It occurred to me that a mobile crane has so many moving parts you could simulate: extending, lifting, bending, rotating the arm, and opening the claw. It would be a fun project to do, and I know I was the right person to do it. Setting up and tweaking the physics parameters is hard, and takes a lot of practice in getting right. I had that practice from the open source games that I had made.

The app store is full of simple one-touch games. I was doing a full blown crane simulation with 10 different controls, so that does set you apart. But marketing was not in my mind: I was doing 'technology push' development, not 'demand-pull' development. This turned out to be the key.

How did you get started?

Literally, quit my job, and start coding. The first thing I needed was programmable shaders which I had not done before. I knew how to make iOS apps, and I knew OpenGL and 3D graphics, but programmable shaders for iPhone were new to me. So I started with writing code to support that, and then started building up my crane simulation. The nice thing about simulations: they are fun even before the game play is in place! It is a good motivator: just playing with the simulation is already rewarding, so adding a game concept with goals, game levels and such is a relatively small step.

If you started it while you were working, how did you find the time to do both, was it difficult?

I learned how to develop for iOS in my spare time, during evenings or weekends. This normally would have been very difficult considering the overtime you typically do at your regular job. But it helps if it is a hobby, and it is the favorite thing you do. Making games is fun, period. My costs were only a few thousand dollars, to get a Macintosh and an iPhone. The main cost is your own time. My recommendation is to start a side project if you really enjoy doing it. If it is for the potential money, and you have to force yourself to work on your project without really enjoying it, you are probably doomed.

What is your business model?

Being a one man team means that I had to do my own artwork. This is of course not ideal: the visuals will not match what big studios are able to deliver. Paid iPhone games have to look the part. So I decided to make my game free for the first few levels, and to play all the levels, the player have to make a purchase. A lot of people would be trying it out, and if I can get them hooked, the money is in the pocket.

When you launched, how long was it until you saw your first sale, and where did it come from?

OK, this is actually quite a funny story. First of all, I did not do any marketing. I did announce my project on indiegamer.com which is a forum for developers like me. One of the responses to my announcement of 'the little crane that could' was pretty clear cut: the game will suck. A developer explained that they coded pretty much the exact same game at a studio he worked at, and when they focus-tested the game, the focus group told them it was not fun. The focus group also said: there is nothing you can do to make it fun.

Wow... reading that... It made me doubt my own project. I already had a nagging feeling that I had made a game for myself: a game that I would like to play, and not necessarily something others would enjoy. Could the other developer from the forums be right? Probably.

So, then the release day came: day 1 was disappointing. On the first day on the app store, I actually managed to convert one user from a free player into a paying customer. One single sale, I was downtrodden. However, each day, sales were going up tenfold. On my best day, I actually sold 6000 copies on a single day. My game was #1 in the UK and was raking in good money.

I had a hit on my hand, but this was pretty much despite my marketing efforts: I did not do any. I had actually released an unpolished, half-finished game with flaws. The core game play was nevertheless so good, that the game still took off, contrary to popular belief, without marketing. As a matter of fact: my game was largely ignored by the press. Then came the curating phase: I did 13 updates or so, in just as many months, forever improving my game.

What parts of development do you do and what parts do you hire out?

I did everything myself, except for the music, which I bought from a composer advertising his services on the indiegamer forums. For the 3D artwork I used Wings3D modeller. For the 2D artwork I used inkscape. All the code is done in Objective-C using Xcode. By not using any professionals for art, I had to rely on its one main strength: the physics are top notch, second to none, and operating a crane in 3D is fun to do.

How are you marketing it?

As I previously mentioned: one sale the first day, but then it exploded. The first million downloads were only a matter of weeks. And a small but healthy fraction of those free downloads lead to sales. My biggest pay out was in the first month, with a \$58K royalty payment by Apple. Every month after that, the royalties never went below my salary as an employee, often a lot higher. It did not make me a millionaire, but I am now making a lot more money than I ever was as an employee.

Now, a year later, sales are still very nice. It never got back to the levels of the launch month, but more than enough to live off.

The biggest success was in UK where it was #1 free iPhone app and #1 free iPad app. Also, the game was ranked #1 iPad app in quite a few countries. See here: <http://stolk.org/tlctc/palmares.html>

A year after the release, I did try to advertise my game using AdMob. I have not seen any noticeable effects from that. I did actively engage in conversation with my players via Touch Arcade forums, twitter and email.

What has been your biggest surprise so far?

Maybe not so much a surprise for me, but it was contrary to popular belief: the 'race to the bottom' does not have to be. Twice I doubled the price of my full game. First from .99 to 1.99 and I noticed that even though the number of sales dropped, it was not by much. The same thing happened when going from 1.99 to 3.99. Once your game is already free, the barrier to entry is low, and you can price your full game higher.

What has been your biggest mistake?

I added advertisements to the free version, (which are disabled when the user purchases the full game). As advertisement framework, I chose Apple's iAd. This was a bad choice. At the height of popularity, *the little crane that could* was generating 1.9M ad requests in a single day to Apple's servers. iAd was in its infancy, and only managed to fill 3% of my ad requests with an actual ad. Today, iAd's performance is not much better, and only 20% of ad requests are honored with an actual ad. Those 1.9M ad requests that day, netted me \$600,- which I think would have been more had I used another, more mature ad network.

What tracking tools do you use?

I did not do any tracking, but I did keep iAd enabled as the ad-request volume during a day accurately predicts the sales, which are reported the next day by Apple.

How is your project doing now, and what are your plans for the future?

Still going strong. Every now and then, the game will top the charts and get to #1 in small countries. The big money maker is the US market though, and the game never really broke through in that market.

About the future: after *the little crane that could*, I made another game, *Hover Biker*. This time I had a decent art budget thanks to my sales. So I hired artists, and put out a new game, which I thought looked so much better, and could have broader appeal as well: everybody likes flying motorcycles! That game saw little sales though. This is kind of worrisome: yes, I managed to make a hit game, but I struggled to repeat my success. For now, repeat has not been necessary, as the little crane is still selling well. But what if it stops doing that? How can I repeat my success?

So for now, I am 'condemned' to focus on my one hit game. I will be releasing a level editor for it soon. I think user-designed levels can bring the game to a whole new level.

What advice would you give to a developer that wants to work on their own side project?

Use a side project to build up your confidence. Prove to yourself and the world that you can generate revenue. How much, does not really matter. If by working in weekends, you can create a \$1000 monthly income, then by cranking up the effort to full time, making a living out of it should be easy. The scaling in revenue by going from part time to full time is often better than linear. In my case I put in three times the hours, and started seeing 10 to 20 times the revenue. Just get those first sales going, so you can hit the ground running. When I went independent, I already knew how to publish iOS titles and make money from them. That is a big bonus when starting on your own.

In short: use that side project to make your first sale. The other sales will follow naturally.

Also, use the side project as personal training. After doing a full cycle from conception to launch and beyond, you will have the complete picture of all that is necessary for a project to be completed. There will be fewer surprises when you take the plunge and go full time with the next project.

And maybe the most important piece of advice: don't try to gauge the market. Make what YOU would like to use, not what you THINK the market wants to use. I tried to do the latter, and created *Jump Daisy Jump*, which I thought was the simple dumbed down game that the market would gobble up. Well, it bombed big time. Only when I made the game that I thought was too much tailored after my own taste, I got my success. It turned out that own personal taste had a much broader appeal than I thought it would have. Do you think Notch of Minecraft fame was convinced that tens of millions would be playing his non-conformist, pretty 'odd' game? Sure, it was fun for him and his nerdy friends. But tens of millions? It turned out it was. So go where your passion leads you, not where some analyst thinks the money is.

Artsy Editor



Browser

Idea: Create a simple word press plugin editor for writers.

Project Maker(s): Stephen Ou

Website: <http://www.artsyeditor.com/>



Tell us a little about how you got started in web development and software?

I got started when I was 13. It was a funny story because I was trying to find some extra curriculum activities to do. I stumbled upon some tech blogs which drew me in due to the fact that I was fascinated by how technology in general can make people's lives easier, and I decided to be a part of that.

What were you doing before you built Artsyeditor?

I built 4 products before Artsy Editor. They all had been quite popular side projects in their perspective niches that received quite of bit of media coverage and social media love.

OhBoard: a Chrome whiteboard application, <http://ohboard.com>

iTunes Instant: a fast web-based iTunes search engine, <http://labs.stephenou.com/itunes>

TwtRoulette: a mashup that lets you view other twitter home timeline, <http://twtroulette.com>

OneExtraLap: a social quizzing platform, <http://oneextralap.com>

What was your big differentiator?

I would say the biggest differentiator is our design. The way you format your posts and manipulate images is as easy as you can get in a web interface. It saves writers a lot of time. The story behind-the-scene is quite worth noting: before I launched it, I did a complete customer validation process with 200+ people, and the results successfully proved this is something people were willing to pay for. Therefore, without a doubt, I can build the plugin worry-free from having no customers.

If you started Artsyeditor while you were working, how did you find the time to do both, was it difficult?

I have free time at nights and weekends, and the motivation from my customer validation really pushed me working all the time. If you are planning to start a side-project, I recommend doing customer validation before writing code so you have people behind your back to motivate and encourage you virtually while you are working by yourself.

What is your business model?

Simply charge people for the product, plain and simple, just like what traditional business had been doing for decades. I think it's working well because our pricing makes sense on their first impression and they aren't worried about all the extra fees and charges after the initial purchase. Fanciness on pricing isn't an advantage here. Complex pricing is something that small side projects should avoid.

So when you launched, how long was it until you saw your first sale, and where did it come from?

I literally had a sale within a minute I announced the news to our email list. Why? Because a group of people (approximately 800 people) were already waiting to purchase the plugin as early as my initial validation interview. They were aware of what Artsy Editor was all about and what problem I was solving. So they could make the decision quick as soon as they tried out our working demo.

I am currently trying out different free/paid, viral/non-viral marketing strategies right now and see which one converts the most sales from minimum costs. But so far, it had been all word-of-mouth.

I completed all the coding by myself so the starting cost is nearly zero. I only had to pay for the website theme, domain name, hosting server, backup provider, and several tracking services such as Clicky, Verify and Visual Website Optimizer.

Can you walk us through the tools you used to build the initial version of Artsyeditor?

It was a quite straightforward process. It took 2 months of nights and weekends because I want to deliver to those willing-to-pay customers as fast as I can. And I accomplish it by cutting unnecessary features and keeping it simple. It is a WordPress plugin so all the backend code is in PHP, and I use jQuery in the front end which is also where most of the magic happens.

How did you deal with technical issues?

Don't piss off your customers. When you receive a bug report from a customer, reply with a quick note saying you are on it before jumping in to fix the bug. That can make customer satisfied with the response time and have an understanding of what's going on. Try to fix the bug as soon as possible as you don't want to give your paid customers a non-working version as soon as they bought it. If anything is taking too long, remember to keep your customers updated with the latest information.

What parts of development do you do and what parts do you hire out?

I did all design and development myself, but for the copywriting and some marketing, I hired some professionals on Elance.

How was your growth during the first three days, three weeks of launching?

We turned profitable since day 1. Within one month, our total sales are into four figures. And at the time of this interview, we are very close to reaching 300 customers.

What has been your biggest lesson so far?

Marketing is not a onetime effort, you have to keep trying different strategies thoroughly until figuring out which one works.

What tracking tools do you use?

Visual Website Optimizer and Clicky. It's actually quite easy to integrate one into the other since they both allow custom code configuration. I can simply add a short snippet in VWO code to tell Clicky which version of the homepage had generated the sale and how much is it.

How is your project doing now, and what are your plans for the future?

Artsy Editor is selling really well; we've been profitable since day 1. This is the first time that I actually made a sizable income from a product of my own, so I still have a lot of plans to grow Artsy Editor to a bigger product and continue to improve more people's WordPress writing experience. In the next few weeks, I will be trying out guest-blogging as a marketing strategy and see if it will have any noticeable increase on our brand awareness and product sales.

What advice would you give to a developer that is just about to begin a side project?

Focus on solving a real problem for people. There are so many misconceptions out there that you should always buy a domain and start writing code right away when you have an idea. From my experience, that never works. The process of idea validation is crucial to the idea's future success. If nobody is interested in your idea, there is no purpose to build it anyway. So the first thing you should do when you have an idea is to ask people around you, see if this is a problem needs to be solved. If it is, great, start prototyping your idea based on the problem (don't start adding needless features). If it isn't, it's not too late to stop.

Debt Snowball+



iPhone

Idea: Create an app that harnesses the debt snowball method.

Project Maker(s): Derek Clark

Download App: <http://itunes.apple.com/us/app/debt-snowball+/id516229756?mt=8>



Tell us about how you got started in web development and software.

I started out learning iPhone app development on the side while

I was still working full time for a software firm. At that point it was just as a hobby and learning experience for me. This turned into me doing a few apps for the company I was working for which was great. This let me work on learning more and more about app development during my normal work day which was a big plus. Eventually I started working on apps on the side as a way to make a side income and hopefully start my own business.

What were you doing before you decided to build a side project, and how did it come about?

About 5 months before I ended up quitting my job I decided I was going to work hard and really try to go out on my own. After doing a few smaller utility apps I found that I had a real need for an app that didn't exist yet on the store. I decided I wanted to fill that gap, even if it was just for me. What became our envelope budgeting software named Ez Budget was the first really big side project I did, and it was also the basis for eventually quitting my day job.

Why did you decide to launch this?

After launching our budgeting software the next logical step for us was to add a debt tracking app. We used the debt snowball method to get out of debt and we wanted to help other people do the same. There were other apps on the market that do this, but there were things about each of them that we didn't like. We wanted to make something as easy as possible to use that also looked great. We also wanted to include a few features that we haven't seen in other similar apps.

Once you decided to do this, how did you literally get started?

I worked with a designer for quite a few weeks going back and forth on requirements, workflow, layouts, and other design questions. The design and architecture in the beginning is sometimes the longest part of an app. Once we got the design the way we liked it we went to work on actually putting the app together.

If you started it while you were working, how did you find the time to do both?

When I was still working it was incredibly hard to find the time for everything I wanted to get done. For the last few months I was working full time I would come and work on my apps every night and pretty much all weekend. It was almost like working 2 jobs full time. The costs of app development are thankfully very low. The only thing you really have to have (other than a computer) is the \$99 developer fee from Apple. Other than that there really aren't any costs.

I would probably not recommend working quite as much as I did to someone else, but you do have to be really good about time management. Find things to cut out if they aren't that important. Instead of just coming home and watching TV, I picked out about 2 shows a week I really wanted to watch. When those were on I was relaxing and wasn't working. I didn't just sit on the couch and channel surf though. If I didn't have something important to do that was a good time to be getting some work done on my app.

What is your business model?

I decided from the beginning that I wasn't going to plan for having a "hit" app. Getting to #1 on the app store would be really awesome, but it takes a tremendous amount of luck when you consider there are over 600,000 apps on the store. I decided I was going to try to hit niche markets that meet a need. Apps that people are actively searching for. The key part about this is that it makes sales a bit more consistent. If you make a game, it is sort of a lottery. You might hit it big, but you might not ever sell one after you are off the new release list. When I started that was too big of a risk for me. I wanted to make sure I could consistently sell some apps every day, even if there was no chance for these apps to make it to #1.

This model has served me well. Within 6 months of quitting I had replaced my old salary without having any single app be a hit. Now that I've had an app that has done pretty well on the store (Debt Snowball+ got to #125 overall and #2 in Finance) I'm even more convinced this was a good strategy for me. This still isn't the type of app that could ever overtake Angry Birds at the top, but it is something that people need and search for and it was able to get pretty high in the rankings sell a significant number of apps.

Can you walk us through the tools you used to build the initial version?

Objective-C is the language used for building iOS apps. The 1.0 release of our first big app took about 6 weeks. The latest app was a few months from the start of design to the 1.0 release on the store.

What parts of development do you do and what parts do you hire out?

I do all of the coding and I hire a designer to do all the graphics for me.

How are you marketing it?

Thus far growth in the business has been very steady, with upticks coming with each new app we've released. Because of the strategy of going after niche markets most of the apps have continued to have consistent sales over time. Each new app has brought us to a new plateau of revenue. As for the latest and most successful app it is too early to tell. We had several major blogs write good reviews for Debt Snowball+ within the first couple of days and that had a pretty big effect on our initial sales. Once that traffic died down sales have stayed pretty high, but it's too early to tell where they will settle in at long term.

What has been your biggest surprise so far?

The consistency of sales over time. They fluctuate quite a bit from day to day, but for the most part sales in a given week or month are pretty consistent. Apps that have been on the store a long time still sell in the same range pretty much every day.

What has been your biggest mistake?

Taking an app off the store to get an update with a bug fix in it. It sounded like a good idea at the time, but it ended up losing pretty much all of its search rankings. It would have been much better to leave it on the store and deal with the support requests for a week.

What tracking tools do you use?

I use AppFigures.com for tracking all of my sales, rankings, and reviews. It allows me to quickly see how things are going, and respond to any changes (such as increased sales in a different country) very easily.

How is your project doing now, and what are your plans for the future?

Things are going well. I'm coming up on 1 year on my own now and I couldn't be happier with how things have gone so far. We have lots of plans for updates to our existing apps as well as several new projects we hope to get out in the next year. It should be exciting.

What advice would you give to a developer that wants to work on their own side project?

Just keep at it. If the first thing you try isn't a huge success don't let it get to you. If you think it has potential keep working at it and making great updates. If it doesn't look good, move on to the next project and give it a shot. The other advice I'd give is to find a great designer. If you aren't a good graphic designer don't try to do it yourself. The best thing you can do to increase your sales is to have an awesome icon and a slick interface for the app. It's definitely worth the expense to pay for quality design.

Bidsketch



Browser

Idea: Create a simple tool to create web design proposals.
Project Maker(s): Ruben Gamez
Website: <http://www.bidsketch.com/>



Tell us a little about yourself, how you got started with web design, and your background in building side projects before Bidsketch.

I've been in web design and development for over 10 years now. I'm self-taught and early on web design was just a hobby that I didn't expect to earn money from. Learning simple HTML was mind bending for me back then. It was so difficult and everyone kept saying how easy it was; there was no way I could see myself advancing very far but I kept at it anyways.

Eventually, I somehow got the hang of things and started learning more difficult languages until I actually became decent enough to fool people into thinking that I was a real web designer. I spent every waking hour writing code. Then people started asking me to write code for them. It was nuts. And I loved it.

So around this time I decided that I'd try getting paid for it by getting a full time web development job. With no real experience I focused on a less popular programming language called ColdFusion, got certified and created a website where I would teach others how to get certified.

That strategy paid off and I was able to get a (low paying) entry-level job as a ColdFusion developer.

Once I had a job I was exposed to all sorts of neat web apps that I saw people paying for. My company was paying thousands of dollars for 3rd party products. Some of them complex but most seemed relatively straight forward: Content, project and document management systems. That sort of thing. It was easy money (or so I thought).

So I decided to build my own products and sell them to people. But I would start building and never finish. Every single product I tried to build followed that pattern. After the initial excitement of starting on a new product wore off, I would get bored and start working on another idea. This happened at least five times.

How did you come up with the idea for Bidsketch?

I was working full time when I came up with the idea for Bidsketch. Not only was I working full time but I was also working on a side project that I was starting to get bored of (since this was my pattern). Right around that time a friend of mine asked me about web design proposals. He had never written one and a company wanted him to submit a proposal to bid on a web design project. I searched for templates online and stumbled on a product that allowed web designers to create proposals. Unfortunately, it was an old-school downloadable software plugin for MS Word. I looked for similar products and didn't find anything web based. This led me to do some more research into the market for that kind of product.

I had been reading Eric Sink's blog by then so I knew that marketing was important and I needed to make sure that this was something people wanted and would pay for, before starting to write code. I researched the competition, which told me that web designers were paying for that kind of software. I then used Google's AdWords keyword tool to see how many people were searching for related terms. The results were encouraging and I decided to quit development on the test case management system and start working on Bidsketch. (Sound familiar?)

This time around I wanted to approach things differently and started with the marketing side of things. I put up a landing page and blog, and then started writing posts targeting specific keywords related to the product. The sole purpose of my landing page was to collect emails, while my blog was supposed to generate traffic. Shortly after putting up the blog I started commenting in web design forums and attempting to answer the businessy type questions. Being a developer I couldn't hold off on the code for very long so I also started writing code. But I didn't write Bidsketch code, I wrote code for a free mini-app called Estimate Helper as a way to learn how to program in Grails and help generate some traffic.

After a month and a half working on Estimate Helper it launched.

Bidsketch, Early Mockup

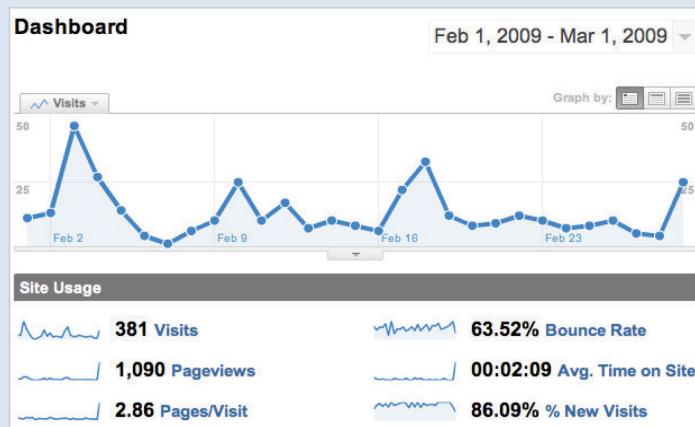
Bidsketch, Initial Email Collection Page

Bob Walsh even announced the release on his blog: <http://47hats.com/2009/02/microisv-digest-19/>

Ruben Gamez, Earthlingworks, has released a part of his larger Overture (web based proposal software for small businesses or freelancers) as a ready to use now app, Estimate Helper. A very helpful tool for making an estimate that's realistic. (via email).

Note that I was calling my product “Overture” at this time and later changed the name to Bidsketch to avoid getting the crap sued out of me by Yahoo.

I was getting an underwhelming 300 to 400 visits per month to my blog at that time:



Estimate Helper got a small amount of some of this traffic and a few visits from the announcement post but eventually “slowed” down to about one visit per page so I took it off line. I learned to program in

Grails but failed to generate traffic with it. At this time I was writing blog posts, posting in forums, writing code and doing design while working full time. I had no free time; this went on for another couple of months. I was exhausted and was progress was extremely slow. I was getting burned out.

So you decided to build it, was it just you or did you work with others?

About two months into building Bidsketch I was only ¼ of the way there. I couldn't see myself doing everything that I was doing for another six months. Also, my forum posting was getting me nowhere. People just weren't interested in there and I was having serious doubts on whether people would pay for the product I was building. I emailed one of my favorite bloggers at the time to get his opinion: Patrick McKenzie. I asked him what if I might be going after a niche that was too small. He emailed back with a very thorough and helpful response saying that he thought it was a great niche with lots of potential. He also gave several blog post ideas and suggested I focus on putting out great content. Patrick's reply was super helpful and helped me refocus on the right things. Shortly after that exchange with Patrick, I struck up a conversation with Rob Walling who was launching a learning environment and community for startups called Micropreneur Academy. I was first inline when he launched and was blown away by the fantastic information in there. This is where I realized that I didn't need to do everything else and I could outsource things. I stopped development in Grails and started over in Ruby on Rails to make outsourcing easier. I hired a designer to do the design, a developer to help me code and a virtual assistant to handle anything that would take time away from working on my product. So yeah, I went a little nuts outsourcing stuff. Keep in mind that I didn't have a lot of money so I used services like Elance and Guru to find help I could afford. Admittedly, finding quality help wasn't easy but the effort was totally worth it. I went from nothing to beta in four months and was able to focus on marketing at the same time. Seeing so much progress in such a short amount of time was amazing. I'm convinced it's the main reason why I was able to launch a product and focus on marketing while working a full time job at the same time.

Is BidSketch your full time project now?

Yeah, Bidsketch is full time for me now. I launched in October of 2009 and was able to quit my job in March of 2011, so about a year and a half.

So you built this thing, what happened next?

While in beta one of my beta testers sent me an email to a web based competitor. Their design was amazing and they had gotten a write-up by 37signals. I didn't have any competition before I started building Bidsketch and right before I launch I run into this amazing looking product. I was floored. It felt like I had just been punched in the stomach. But a few days later I stumbled onto their listing on an auction site where they listed their business for sale. Excited I emailed one of the founders and was able to work out a deal to acquire them for terms I could afford. It helped that the co-founder I was talking to liked what I was doing with Bidsketch. I decided to use their amazing design for Bidsketch and migrate their users over but that would have to wait until after the launch, so after the acquisition I changed gears again and focused on launching Bidsketch. I had a special private launch for people on my mailing list. This was a five-day period where people on my list could sign up to a free plan or a discounted premium plan. There were only 300 people on my email list but 19 signed up for the free plan and 21 signed up for the paid. Needless to say it was pretty exciting considering that I was worried that no one would sign up.

What type of business model did you decide to go with?

There was a special private launch with two plans: free and premium. This was the same for the public launch, only I raised the price on the premium plan. By this time I was getting about two thousand visits a month from the blog. The month I launched I had just over 11 thousand visitors because a couple of designer blogs posted reviews. I had gone out and sent cold emails to over 30 design related blogs but was only able to get three of them to write anything. It was a tough process because I wrote custom emails for each blog instead of using a template. Even with all of this traffic, I soon started to average about two paid signups a week which was discouraging because I also had several paid signups canceling before their trial was over. Revenue growth was slow or non-existent for about six weeks after launching. I tried a/b testing all sorts of things to try to increase the number of paid signups but saw minimal results. Getting feedback from people that were canceling was also a problem – no one would reply back to my feedback emails, even when I customized and sent each one manually. There were three plans now: free, basic and premium. Two paid plans but 99% of people were signing up for the free

plan so I experimented by getting rid of the free plan. Immediate results were pretty encouraging as I saw an 8x increase in paid signups and big increase in revenue. This became a permanent change and I no longer offer a free plan.

How many versions have you had since the initial launch?

It's difficult to talk about versions because it's been such an iterative process that it slowly has evolved into a different product than it was when I first launched. I prefer smaller releases to reduce risk and the keep the number of bugs low. Major milestones within the product:

1. Added a templating system that allows anyone that knows HTML/CSS to create their own proposal templates for use within Bidsketch.
2. Major rewrite of how the content of the proposals was presented when editing. This helped with speed which was a big problem for the first couple of months.
3. Updated the design and added the ability to use custom domains + brand the entire app.
4. Introduction of the proposal designer feature which allows lots of flexibility when customizing the design of built in proposal templates.

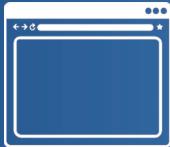
Has it done as well as you thought it would?

My original intention was to learn how to launch and run a business and earn a bit of revenue on the side. I thought of it as a step towards independence and away from working for the man. That said, I didn't expect to get there with this product alone so the success I've seen has been a surprise. Currently, I receive about 13k to 16k unique visitors a month and have around a thousand paid accounts. It's growing at an exciting pace and I'm working on growth at the moment. I'm also starting to focus on educating my market in ways to grow their business through sales. It's insanely important that I do what I can to ensure their success.

What advice would you give to someone about to build their own side project?

1. You don't have to do everything yourself. Find a way to outsource something -- anything. It's amazing what it'll do for your momentum.
2. Start marketing before you start building your product. Devote at least a quarter of your time (hopefully more) to marketing while you build. Outsourcing helps free your time for this.
3. Collect emails from potential customers. Put up a landing page if you don't have one. If you do, find ways to get traffic to your landing page and give people something for signing up to your list. Incentives are a big deal and they work.
4. Keep your beta testing group small. Better yet, don't call it beta; call it pre-launch or something exclusive sounding.
5. Don't just go with freemium because everyone is doing it. Experiment with paid only it can mean a huge increase in revenue. Make sure you couple this with requiring a credit card upfront, even if you're doing a trial. Those two things combined are very effective.
6. Talk to people that are doing the same thing that you're doing. It's important to connect with others that understand and can help you during your low points because you'll need it.
7. Ask a lot of questions. Email people and ask them stuff. There are a lot of different ways to build a successful product, learn as much as you can from other people's experiences.
8. Do what you can to launch faster (cut features, outsource, etc.)

ParkWhiz

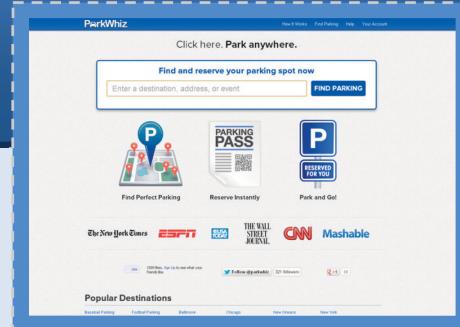


Browser

Idea: Create a web based system to reserve parking spots.

Project Maker(s): Jon Thornton

Website: <http://www.parkwhiz.com/>



Tell us a little about how you got started in web development and software.

I got started with web development on my own when I was 12 or 13 years old. I discovered that you could click "view source" to see how websites were put together, and that's how I learned for the first year or so. I'd find web sites I thought were impressive, then I'd read the code to figure out how they worked. This was before minification or significant server-side coding - static HTML and FTP was how sites were built then.

From there, I took any project I could get my hands on. I built sites for friends, community organizations, local businesses. An internship at IBM showed me how to manage bigger projects, but I was mostly self-taught.

What were you doing before you decided to build a side project, and how did it come about?

I had just finished my undergrad degree, and had the summer off before starting grad school. Several classmates from undergrad announced plans to commercialize their thesis projects, and I thought trying to launch a business seemed like a good way to spend the summer. My thesis project was on using technology to make parking easier, and I had the good fortune of being introduced to the person who would become my cofounder by my professor.

When classes started in the fall, I decided it was more cost-effective to get an education by seeing this project through (free) instead of going to grad school (\$50k/year).

Why did you decide to launch this?

The initial concept for ParkWhiz was to build a parking guidance system: a set of sensors and displays that would tell drivers where to park. These systems existed for large garages, mainly in Europe, and we wanted to make them cheap enough to be commonplace. I had designed a sensor for my thesis project, and ParkWhiz was going to manufacture and install these sensors in garages and parking meters in the US.

Turns out that starting a hardware company is not easy, especially if you have no experience running a business and you're working other jobs to support yourself. We tried a couple different ideas but nothing stuck. OpenTable was just getting big at the time, and we realized that a web-based system to reserve parking ahead of time might work.

Nothing like ParkWhiz existed when we launched in 2007. In fact, it took the market a few years to understand what we were building, and things didn't really catch on until 2009.

Once you decided to do this, how did you get started?

I had the summer off, so I was able to dive in full time to get things started. I hacked away on a product - sensor designs at first, then the web-based reservation system, while my partner worked on a business

plan and tried to find potential customers. The first summer was key. We worked so hard, it was tough to just give up when the sensor idea didn't pan out. We just kept going.

If you started it while you were working, how did you find the time to do both, was it difficult?

I was lucky enough to start the project full-time, so when it came time to get a "real" job to support myself, my priorities were already focused on ParkWhiz. I was working on ParkWhiz nights and weekends, but it was my day jobs that felt like moonlighting. I did freelance work for ad agencies, worked for a tech startup, took random consulting jobs, anything that would provide me the flexibility to continue working on ParkWhiz.

Aside from salaries, web-based businesses typically have few costs. Server hosting cost about \$50/month, and that was the biggest expense. My cofounder invested a small amount of money in the company and won a few business pitch competitions, which gave us our initial funding of \$40,000.

What is your business model?

When a driver reserves a parking spot through ParkWhiz, they pay via credit card in advance, and then ParkWhiz pays the parking owner for the spot. In between, ParkWhiz takes a commission on the transaction.

It took us over a year after launching to make more than \$1,000 in a single month. The site now regularly takes in far more than that every day.

When you launched, how long was it until you saw your first sale, and where did it come from?

ParkWhiz launched in April 2007, partnering with a small parking company in Chicago that had 4 garages. We didn't get our first sale until August. That was a rough 4 months, but we stuck with it.

I still don't know how that first customer found us. We had poor SEO, little word of mouth, and social media didn't exist at the time.

Now we market ParkWhiz through a combination of SEO, AdWords, and word-of-mouth thanks to our awesome customer service.

Can you walk us through the tools you used to build the initial version?

ParkWhiz has evolved quite a bit since it launched, but the core tools have remained the same - PHP, MySQL, HTML, Javascript, CSS. I set up Apache on my laptop to build the site, and then used FTP to upload the files to our cheap-o 256MB virtual server.

It took about 3 months to build that initial version. The site has been rewritten many times over by now, but it's always been gradual updates - no relaunch or distinct versions.

The look and feel of the site is surprisingly unchanged from the first version that launched in 2007.

What parts of development do you do and what parts do you hire out?

I wrote every single line of code and managed the servers for ParkWhiz from inception til August 2011. We wanted to build an iPhone app and I didn't have time to pick up Objective-C while still maintaining the site, so we hired a local contractor to build the app. It cost about \$5,000.

By that point the business had started taking off, so I started looking for web developers to work on ParkWhiz full-time. ParkWhiz finally hired its first non-founder developer in April 2012.

How are you marketing it?

ParkWhiz has grown at a remarkably steady rate of 300% per year since we launched in 2007, and continues to do so today. SEO drove growth of the business in the beginning, but we're working harder to use word-of-mouth as our customer base grows.

What has been your biggest surprise so far?

The first time we used AdWords was a real eye opener. The first month we spent \$50 on ads, which resulted in \$100 in revenue. The next month we tried spending \$100 and made \$200. It was like discovering a money machine.

What has been your biggest mistake?

My partner and I tried to raise VC money for the project in 2006. The VC world was still hurting from the dot-com crash, and neither of us had a track record or connections - we were cold-calling potential investors. We finally convinced someone to give us a term sheet for \$1M, and spent \$10,000 of the company's money - 25% of our total funds - having a lawyer review it.

Soon after we discovered that the VC partner we had been talking with offered the term sheet without the full support of the firm. The deal fell through and ParkWhiz was out ten grand.

What tracking tools do you use?

We use Google Analytics for tracking and a/b testing. I confess that I didn't really spend a ton of time understanding the analytics while I was building ParkWhiz. As long as the charts were going up and to the right, I'd rather be coding up new features.

How is your project doing now, and what are your plans for the future?

The project is doing well! ParkWhiz has offices in Chicago and New York City and employs 12 people as of this writing. And we're hiring!

What advice would you give to a developer that wants to work on their own side project?

Don't quit. Find something you care about that doesn't depend on help from others, and just keep working. As long as you're working, you're learning about the problem you're trying to solve. Stick with it long enough, and one day you'll realize that you've become an expert.

It helps to have a good idea, but the reason ParkWhiz has been successful is that we just kept working.

Monster coloring Book

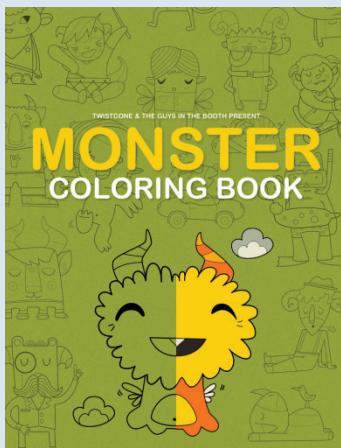


Idea: Create an iPhone/iPad monster-themed coloring book app.

Project Maker(s): Christopher Taylor

Download App: <http://itunes.apple.com/us/app/monster-coloring-book/id435676573?mt=8>

iPhone / iPad



So tell us a little about yourself, how you got started with web design, and your background in building side projects before Monster Coloring Book.

I've been involved in web design and development since 2004. (I guess that's what happens when you study philosophy and then opt not to go to law school at the last minute.)

I'm self-taught but I've leaned heavily on my partner (who studied CS in school) and sites like Stackoverflow.com.

From 2004 to 2008, I primarily worked as an interactive producer, planner and strategist for a variety of ad agencies servicing some big corporate brands (Delta Airlines, Boost Mobile, Amtrak, Jockey, General Mills, etc). Part of my interest in hands-on development work came from being repeatedly told by our agency dev teams that the projects we were contemplating were impossible, too expensive, would take too long, etc. The other half of my interest is courtesy of the team at Cupertino. When the first iPhone launched I felt like I had stepped into a sci-fi movie. I just **knew** I needed to create things for it. The platform and the device were too compelling.

And so in November of 2008 I began the long journey to understand the often byzantine workings of Objective-C. I started learning code in every free moment I could find. In August of 2009, I quit my six-figure job as an interactive planner to take a part-time gig doing similar work (4 days a week) that would still provide some income, but would allow more free time to dedicate to dev work. In August of 2010, I quit that part-time job and dove headfirst into a mix of freelance consulting and independent development work. The first real app-related breakthrough came in November of 2010 when we released Build A Train. [For more details on Build A Train see: <http://theikidsblog.com/blog/2010/11/30/arr-there-be-pirates-in-these-parts/>]

That project was successful enough that it gave us hope that we could make a run at real app store success. So we decided to really invest some time in **Monster Coloring Book** and see if we could create a winner. Because we were relying on a mix of consulting income and app store revenue to survive, investing the amount of time it takes to produce a truly polished product was a risk – but it was one that we embraced.

How did you come up with the idea for Monster Coloring Book?

Almost all of our app development efforts have a kid-focused slant to them, for two main reasons - on a personal note, I am the father of two toddlers (with another one on the way!) and my business partner is the father of three. We've come to realize that there are few things more rewarding in life than sharing your work with your kids and having them actually care about it. On a professional note, kids-content on iOS devices is a business opportunity that has been exploding over the last couple years.

So we knew we wanted to do another kids app, and we knew that we wanted it to look great. With this in mind, we partnered with one of our old agency friends, Zara Gonzalez Hoang, knowing that whatever she designed ALWAYS looked great. Zara has an affinity for monsters and the rest of it sort of fell into place from there.

What tools and languages did you use to build it?

We'd been talking about *Monster Coloring Book* since at least September of 2010, but development didn't really begin in earnest until January of 2011. When we scoped out the project, we knew we could pull most of it off, and that there would be a few elements that would require us to, erm, stretch, our skill set. In particular, we found it challenging to find a way to handle flood fill that the processors in iDevices could handle. In the end, we built the whole app ourselves using straight-up Objective-C.

Is Monster Coloring Book your full time project now?

Consumer-facing iOS development is our full-time gig now – and *Monster Coloring Book* is part of that. It's been a 3+ year process to get to this point, with clear transition points: the switch from a full-time job to a part-time job, the switch from a part-time job to consulting, and then (finally!) the switch away from doing any consulting and to just building things.

What happened next?

Because we had invested so much time in the creation of *Monster Coloring Book* (2.5 people at 50 hrs/week for six months with zero outside funding) we were hoping for a strong launch. That didn't happen instantly, but within a week of launch we had been picked up in a couple of key social media hotspots (HN for one..) and we started to receive some accolades. Special thanks to @kn0thing for being the first person with a significant social following to mention us. It made a HUGE difference. We spent a fair bit of time tracking how the word spread, and it's amazing how much of it can be traced back to what a couple of key influencers happen to post.

At the end of our launch month, among a number of other accolades, we'd been named the TheFWA's Mobile App of the Day, added to A Very Short List, named Smashing App's App of the Day and been featured on the front page of the App Store by Apple. We'd also reached the #1 spot in the educational games category in a number of major markets and cracked the top 200 games list in the US.

What type of business model did you decide to go with?

We decided to go with a dual approach for launch – release a premium product (\$2.99) alongside a free lite version with in-app purchase. Our hope was that the premium product would drive trial of the free version and that we could monetize traffic to the free version via in-app purchased (IAP).

If we had to do it over again, I would go with a revised version of a similar approach. The full version of the app would include ALL content, and any IAP in the free version would be done without the use of virtual currency. I also believe we gave away a bit TOO much content in the free version of the app and if we could do it over again, I'd scale that back some.

How many versions have you had since the initial launch?

We updated the full version of the app two weeks after launch. We took out all IAP from the full version, unlocked all content, and dropped the price of the app from \$2.99 to \$1.99. The free version is still in its launch form. We saw minimal impact (sales/app store reviews) from the update despite giving away a lot of bonus content for free. This was disappointing, as we'd seen this approach work before. But because MCB is so deep, there may have been less of a hunger for additional content, at least as compared to some of our other projects.

Has it done as well as you thought it would?

We had a great launch month. Being featured by Apple clearly has a stronger impact on revenue than anything else. Unfortunately, this effect (for us at least) was ephemeral. We enjoyed great performance (close to \$1k/day) for the two weeks we were featured, but once the feature exposure was taken away, we dropped like a rock on the charts – falling from #5 on the Educational Games chart to well below #125. *Monster Coloring Book* continues to perform OK and will continue to be a small stream of revenue for a while, but to be viable as a business we will have to extend significantly beyond this app. We have already begun working on that – see for example: <http://letslearnhowto.com>

As of August 18, 2012, the team behind Monster Coloring Book (me, Victor Johnson & Zara Gonzalez) decided to build on the success of Monster Coloring Book and launch a new company around our kid-focused apps. We applied to Y Combinator last fall and made the interview round. Although they chose not to fund us, we chose to continue and we can currently be found at <http://playtend.com>. Fun what a side project can evolve into, right?

Monster Coloring Book also continues to evolve. It was named to Apple's Best Apps for Kids list last December and has been included on a few of Apple's curated lists throughout 2012. Inspired by this continued exposure, we released an updated version of the app this spring that includes more monsters, colors, patterns and brushes as well as a streamlined interface. We also integrated it with another one of our apps, enabling you to create a jigsaw puzzle from anything that you create in Monster Coloring Book. It's been a fun ride and we are excited to see how the Monsters continue to evolve - they might just find their way into a game soon...

What advice would you give to someone about to build their own side project?

1. Be prepared to give up sleep. Be very sure that you REALLY want to do whatever you are considering.
2. Plan on iterating. Faster to market means faster feedback and the ability to evolve based upon real market feedback. Holing up in a code-cave for an extended period of time can be a bigger risk than you realize – even if the project does well.
3. Don't plan on getting rich off of one (iOS) app. You probably have better odds of striking it big if you play the lottery.
4. Don't spend a lot of time chasing independent reviews – if your product is good, reviewers will find you. If it's not, your request for a review will wind up in a pile next to the other DOZEN people that requested a review that day. Even if you do get a review, keep your expectations for impact small unless it's from a major, major pub with tons of traffic.
5. Target influencers. They can make or break your project if you catch their eye.
6. Don't target influencers by being pesky – do it by building things that are awesome. That's usually what they care about.
7. Don't give up. Ever. (See tip #1)

Visual Website Optimizer



Browser

Idea: Build the world's easiest A/B testing tool.
Project Maker(s): Paras Chopra
Website: <http://www.visualwebsiteoptimizer.com/>



Tell us a little about how you got started in web development and software.

I got fascinated with computers during my 9th grade in school when my dad brought home a PC. At first, it was about games, but I got quickly bored of playing trial/demo versions of games.

After that I started getting my hands wet with Visual Basic 6.0 making small programs like calculator, tic-tac-toe, etc. Simultaneously, I started reading about algorithms on Internet, and stumbled across Neural Networks and other Artificial Intelligence algorithms. So, I started coding these algorithms in Visual Basic and posted them on an online website called Planet-Source-Code.com. That was my first brush with publishing the code I had written on the web, and the feedback I got was positive and encouraging.

One thing led to the other, and I eventually realized I loved software and it would be best if I made my living doing something in it.

What were you doing before you decided to build a side project, and how did it come about?

I started coding when I was in school, and when I started my engineering at college; I came across Paul Graham's articles on how to do a startup. His essays have always been very inspiring, so I thought of doing a startup of my own. So, over the course of four years at college, I tried a bunch of things (and called them startup). None of it worked, but provided great development and business experience.

Why did you decide to launch this?

After college, I started working at a firm as an R&D engineer. But the passion of doing something of my own was still there. So, I made a list of ideas I would like to work on. At that time, I got interested in marketing, algorithms and technology. Online marketing optimization stood out as a field which allowed me to explore my interests. Hence Wingify came into being. I decided to do something that will help other businesses make the most out of their traffic, and optimize their marketing/landing pages for maximum sales and conversions.

Once you decided to do this, how did you get started?

I was still employed elsewhere, but I started working on the concept during weekends and after work hours. My employer was aware of this. I spent a couple of months and developed the first prototype of conversion rate optimization suite, and called it Wingify platform. It didn't work out because it had feature bloat, and I hadn't thought about usability. Based on feedback, I decided to focus on just one feature: A/B testing and nail it completely (with a special focus on usability and ease of use). So, I threw away all code of Wingify platform and spent 2 more months to create first version of Visual Website Optimizer, which was well received by people.

If you started it while you were working, how did you find the time to do both, was it difficult?

Yes, managing full time work and a side project was very difficult. To make matters worse, my daily commute to work was 1.5 hours one side. My schedule was waking up at 6am, reaching office by 9am, leaving office by 6pm and reaching home by 8pm. After having a quick dinner, I would spend 9pm-12pm

working on my side project. Weekends, of course, were all mine, but it meant skimping a bit on socializing. Initial costs were not much. One mistake I did was to get Wingify's website designed and paid \$1000 for it. I could have easily bought a template for \$5 and modified it myself. Hosting costs were minimal too. I had gotten a server for \$20/month or something. So, all in all, it was cheap.

I highly recommend doing side project part time as long as it doesn't get significant traction. Going full time into a side project which you don't know will work or not is very risky. I made a number of mistakes along the way before Visual Website Optimizer clicked. If I were doing it full time, the pressure to succeed would have killed me. Working full time somewhere else gives you cushion of money and a safety net.

What is your business model?

Our business model is a monthly recurring subscription. Early on, I had realized that recurring subscription is the best business model because you acquire customer once, but he goes on to generate revenue for months in future. So even if you stop acquiring new customers one day, your future revenue will be ensured because of existing customers. With one time payment, you have to acquire a new customer every time you need to make an additional dollar.

When you launched, how long was it until you saw your first sale, and where did it come from?

Visual Website Optimizer was in a closed private beta for 6 months, so I had built a list of 1000+ users who were using the product before I launched paid plans. My initial goal for revenue was to make equivalent of my monthly salary, and to much of my surprise I crossed that figure within first week of launching paid plans. In first month, I made 4 times my previous salary. Main methods for marketing was requesting other bloggers to write about the product, and also word of mouth publicity played a good role.

By the way, I had left my full time job about 2 months prior to launching paid plans. This is because 1000+ users generated a lot of support queries and feature requests. With full time job, I found it impossible to support those many users so I quit my job. It was a risky bet to quit job before launching paid plans, in hindsight I should have waited till I had generated some revenue.

Can you walk us through the tools you used to build the initial version?

I used quick and dirty methods for building initial version. I used PHP and jQuery as languages, and for interface I simply bought a theme from Themeforest for \$5 and customized it. First version was very minimal, but it did well whatever it did. In early days, I think website used to get 100 visitors per day or something. Here's a screenshot



Initial version took 2 months of hard part time work, and then it was in beta for 6 more months (I incorporated lot of feedback from users).

What parts of development do you do and what parts do you hire out?

I did everything myself, except for the interface for which I bought a theme. For a long time, I didn't even have a logo. It was just plain 'Visual Website Optimizer' written in some font. From my previous projects, I had realized testing a concept in quick and dirty manner was much more important than trying to perfect things.

How are you marketing it?

Well, what started as a side project is an incorporated company with 10 full time people working on it. Side project is my full time startup now. We signed up 1200+ customers over last 2 years (including brands like Microsoft, AMD, GE, Groupon, etc.) Marketing is still purely organic. We rank well in search engines. Other source of customers is our blog where we regularly publish high quality content and customer case studies.

Recently, we hired our first dedicated sales guy who gives demos and handles sales queries.

What has been your biggest surprise so far?

Biggest surprise is that what started a side project has grown into a fast growing, highly profitable company and has generated value for thousands of businesses worldwide.

What has been your biggest mistake?

It was with earlier Wingify platform where I put 3+ months of efforts coding the platform without getting feedback from even a single user. I was coding in isolation, and only sought feedback once the whole thing had been made. It would have been much wiser to keep potential customers in loop while developing the product.

What tracking tools do you use?

Obviously, we use our own product for A/B testing: Visual Website Optimizer. We also use Google Analytics and Clicky.

How is your project doing now, and what are your plans for the future?

We are a team of 10 people, and plans for future is to incorporate targeting and analytics features into Visual Website Optimizer to make it one stop solution for marketers trying to optimize sales and conversions on their website. We are hiring engineers and sales people and hope to expand the organization.

What advice would you give to a developer that wants to work on their own side project?

Do side projects in quick and dirty manner while always keeping potential users in the loop before and while developing. Also, don't do it full time unless it is generating enough revenue to take care of your living expenses. Do it part time as long as you can. You may need to do a lot of side projects, before you stumble upon the

Living Earth HD

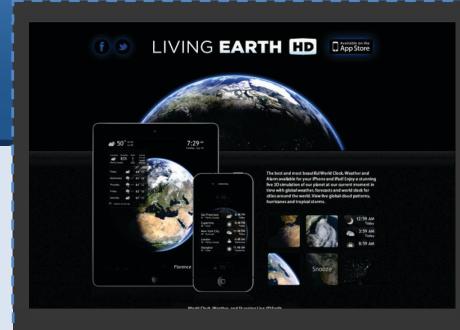


iPhone

Idea: Create a 3D Earth App with weather information.

Project Maker(s): Moshen Chan

Download App: <http://itunes.apple.com/us/app/living-earth-hd-world-clock/id379869627?mt=8>



Tell us a little about how you got started in web development and software.

I've been a software developer for about 14 years. I've never been focused in any particular area of development but always found myself taking on projects or joining startups where I would need to learn new languages, new tools, and new frameworks. Specifically, each new job I took on would be in an area that I didn't have much experience with and I would need to learn everything from the ground up. With the Living Earth HD project I had to learn everything necessary to make an iPhone and iPad app including learning a new language, development environment, and tools.

My first side project was in 2007 while I was working a day job at a startup. It was an action puzzle game called Ryokan for the Sidekick/Hiptop mobile phones. It sold in the worldwide Sidekick app store. Compared to the numbers that you can make on today's mobile platforms the sales were not great. However, the experience opened my eyes to the possibility of being an indie developer; I was able to create something in my spare time, sell it to the world, and make extra income every month. A couple years later I joined a startup that made tools for indie Flash game developers. That gave me the opportunity to interact with many indie game developers who were successful at making game development their full time job. This experience also allowed me to examine what made certain developers and games successful and what made others fail.

What were you doing before you decided to build a side project, and how did it come about?

At the time, I was working at a startup that made tools for indie game developers. At the same time we were acquired, I took on an engineering management role in the company. There were a lot of complex systems and projects that fell within my realm of responsibilities. I was really itching to create something from scratch again that wasn't tied to the complexities of my job and that was all my own. It was the really the appeal of focusing my time after work on the brass tacks of just creating again that got me started.

Why did you decide to launch this?

The current product has evolved quite a bit from my initial concept. I was initially looking for inspiration on what to build by browsing apps and demos on a couple of technology-art blogs. I examined a couple of C++ frameworks that were popular with those that built these kinds of demos and apps such as Processing and Cinder. Cinder had a great project where you could visualize Earthquakes around a 3D Earth. That was when I decided to make a 3D Earth visualizer for real-time Twitter updates. I wanted to make it easy to visualize what parts of the world were awake and the kinds of things they would tweet about.

Halfway through the project I became obsessed with making the best looking Earth rendering that could be done on the power of an iPhone and iPad. I had real-time night/day shading working as well as live cloud systems rendered on the globe. I wanted to create the experience of looking at our planet live from space. By this time, I had spent enough time studying how to calculate the position of sunlight for the time of day and time of year that making a very nice visual world clock was also a product possibility. Because this was simpler than the original concept I decided to release a world clock as a minimal viable product.

Since I had live cloud data rendered on the globe it was an obvious next step to also include weather conditions with the cities. That's how the app eventually became a world clock and weather app.

Once you decided to do this, how did you get started?

I had to learn everything necessary to create this project from scratch. This included learning OpenGL for the 3D engine as well as the framework and tools to develop iPhone and iPad apps. This meant many dozens of hours of making literally no progress on the actual app itself as I fumbled my way in experimenting and learning everything.

If you started it while you were working, how did you find the time to do both, was it difficult?

Both of my mobile app side projects were produced while I had a full-time job. I would come home from work and put four to eight hours or more every night on the project. Both projects came out of personal interests and it was very rewarding to work on something that's just my own. I never felt like I was forcing myself to put the hours in. That said, it's a lot of work. However, putting the hours in is simply what it takes to get something like this done. It's definitely not sustainable in the long run to be working so many hours. It's extremely important to limit the scope of your projects so they're small. I see a lot of people start off a bit too ambitious with their projects and realize later that they've bitten off more than they can chew. Remember, even shipping something small and simple is a huge morale boost that can provide the motivation needed for future features or projects.

What is your business model?

I had no business model. My goal was to create the best product I could. Living Earth HD was intended to be a fun project to enable me to experiment with the Apple App ecosystem. Through my work I had connected with quite a few successful indie game and app developers. I also had a few personal friends who were making their businesses work well on the Apple App store. Their experiences provided a guiding platform early on. However, like anything in technology, the mobile app business is new and changing very fast. I think it's easy to overanalyze a model, or take someone else's experience as convention, and I feel it's important to just experiment as the market dynamics unfold.

When you launched, how long was it until you saw your first sale?

I always thought that if the initial version of the app made enough money to pay for dinner every night then I would have considered it a success. The app made about what a dinner in San Francisco cost immediately after launch and stayed that way for the first month. Because I had a visible and easily accessible feedback button, I received feedback and emails a few days after it was released. It was really exciting when I received the first email about the app from someone that I didn't know. The early feedback that I got from users played a huge role in iterating the app past the very initial simple version.

I've experimented with many ways to boost sales of the app. In order of importance what worked for me are: word of mouth, reviews and blogs, and social media. If I focus on making something that people can depend on and use every day, then word of mouth marketing will do much better than everything else. Beyond that, getting the word out on updates to reviewers and blogs and hoping a big blog writes about you can be very effective. Lastly, I have a button in the app to follow my app Twitter feed as well as easy ways to share a photo of the Earth you take in the app with others through Facebook, Twitter, Tumblr and email. This shows a nice photo as well as a link that takes someone back to the app purchase page.

Can you walk us through the tools you used to build the initial version?

It took three months of working every night and every weekend to build the initial version. Most of this time was spent on learning new things to be able to do the project. Tools used were mostly XCode and Photoshop. The 3D engine is written in C++, OpenGL ES 2.0, and GLSL Shading language. Everything else in the app is written in Objective-C. The weather and maps server that handles all the requests from the app is written in Javascript using Node.js as a server framework. Python is used on the server to update and do image processing on the live maps. The weather and maps server is hosted on Amazon EC2. The maps data is distributed using Level 3 as the Content Delivery Network.

Here's a picture of the initial version when I first started to get the 3D engine working:



The 3D engine evolved and the visuals got better:



Here's a shipping version of the app:



What parts of development do you do and what parts do you hire out

I wrote all of the code myself. However, the product itself was shaped by feedback through many people, both peers and customers. I think it's important to identify where your strengths and weaknesses are and figure out how to cover some of those weaknesses, otherwise it'll really show in the product or business. I've always wanted to be good at UI and design. I've gotten much better, however those weren't my strengths at the time. I think it's important to hire out, or do a work exchange for areas where you could use some help. For example, the first version of the app that had weather forecasts used a design that a friend made for weather features for his own app. If you can do a work exchange it can be a mutually beneficial setup for both people. For the website for Living Earth HD I hired a contract designer and it was well worth the cost.

How are you marketing it?

The app made enough to buy dinner every day for the first month. Every update after that was in response to user feedback and iterating on making the app better. I added more weather details, more cities, iPad support, and improved all of the visuals. Sales grew steadily and about two to three months after the first release I hit the top ten paid rankings in several countries including the UK. This was all purely organic growth. Since then Apple has featured the app in almost every major country. That's always a huge boost in sales however it's only temporary and an app has to be able to sustain organic growth for it to generate a steady income. About six months after the first release I decided to quit my job and go indie full time.

I mentioned my marketing strategies earlier, but I'll mention specifically that I experimented with ads. I had a spreadsheet I used to keep track of ad costs, performance and final ROI. Since the revenue model

was from directly selling a low-cost paid app I could only get a positive ROI from ads if I could buy the ads at an extremely low eCPM. If you can find a cheap enough ad channel this could work, however I feel it's extremely rare if not impossible. Brand advertisers or VC funded startups who are paying to acquire users at higher costs are almost always going to outbid you in ad channels.

What has been your biggest surprise so far?

The amount of feedback that I've gotten has really surprised me. In the end it's a relatively simple world clock and weather app. However, the quality of the rendering of our planet as well as the "live" nature of it inspires people to leave feedback either through the app or by leaving a review. I think a large part of the success is the emotional impact the app can have on people and that's something that I read in the feedback every day. All countries combined, I have over 20,000 reviews and I've given up on reading all of them. Because the in-app feedback is through email and I can respond, I do read all of those. It's great to hear how others around the world are using the app.

What has been your biggest mistake?

It's hard to point to any single big mistake because in software almost no decision is permanent. Almost everything can be reversed or corrected. There can be numerous small mistakes or things that just don't work out that can be quickly fixed before they become big mistakes. I've done my share of botched releases, spotty QA, and load and scale issues for weather and cloud data.

What tracking tools do you use?

I held off on integrating any kind of analytics until recently. My app is feedback driven however it's simple enough to not need to be analytics or data driven. I get feedback through my beta testing group that I manage using TestFlight. I get customer feedback through a very easy to find feedback button in my app. I use Flurry Analytics to track metrics such as usage, installs, sessions, versions and other broad stats. AppViz is used on my Mac to track sales and customer reviews.

How is your project doing now, and what are your plans for the future?

Living Earth HD is still doing well in the App Store. I'm wrapping up some new features that will ship very soon. I've also partnered with a friend to develop the Mac desktop version of the app. It'll have live wallpaper, a screensaver and a drop down world clock and weather interface accessible from the top menu bar. My focus is making it the best world clock and weather app for iPhone, iPad and Mac desktops. After that, who knows? It would be fun to start a completely new project again.

What advice would you give to a developer that wants to work on their own side project?

Figure out the goals of the side project. Is it for fun and to learn something new? Do you hope to be able to make side income on it and hopefully be able to work on it full time? If that's the case it's important to work on something that has both appeal to users and customers as well as yourself and your own personal interests. The former is important because ultimately you want to make a product that can be part of a business, and the latter is important because without a boss and a team to help drive a project to completion it can be easy to lose interest and focus in the project once the going gets rough. Pick a project that's small and can be easily shipped. It's easy for developers to underestimate the amount of work involved, especially when you have to do everything yourself.

Most developers I speak with who want to work on their own side project never ship their project. If you're not accustomed to shipping something on your own then I would make learning how to ship your first goal. Pick a project that seems overly simple and ship it. Then start working on shipping slightly larger projects. This will give you an idea of how long it will realistically take for you to do something on your own, especially if you're balancing work, family, and a social life.

Reach out to others who are working in the same space or have been successful with their own side projects. The communities of indie developers I connect with are always happy to share their experiences and give any relevant feedback and advice on your project. Lastly, just do it. You'll never know where a project will take you until you commit to doing it and getting it out there.

Gumroad



Browser

Idea: Create a web app to sell anything with a link.
Project Maker(s): Sahil Lavingia
Website: <http://www.gumroad.com/>



So tell us a little about yourself, how you got started with web design, your background in building side projects.

I started making things at a pretty young age, and after I built (and sold) my first web "app" I was addicted to making things.

Especially by myself. The feeling of being able to create something people love and find useful by myself is really awesome. I started out designing but slowly learned HTML/CSS, then PHP, then Objective C, then Python, then JavaScript. I'm not amazing at all of them, but I've learned that Stack Overflow and Google are smarter than me and are always there to help.

Your current project is Gumroad, which lets people sell or share files with a single link on twitter, when did you launch it and how did you come up with this idea?

It's not just for Twitter actually. A lot of people use it with Facebook. I had the idea when I wanted to sell a one-off icon I did and realized it was really hard and wasn't going to be worth the effort, especially if no one bought it. I felt like sharing a link was so easy, why wasn't selling stuff? I was working at a startup in Palo Alto. I was able to spend 20-40 hours a week on my side projects when I wanted to. There is always time. It's not hard. I loved working on my own things so taking time out of doing other things to do that was pretty easy. Gumroad will probably soon becomes my full-time gig, yeah. Nope, no costs! Except some sleep.

How do you come up with product ideas?

I like building things to solve my own problems. So even if no one ends up using it, it at least benefits me.

How long did it take you to develop?

It took me a weekend. Probably around 20-25 hours from idea to what it was when I launched it.

What tools did you use to develop Gumroad? Does the current version look like the initial one?

It has changed since we've grown. I used Textmate and App Engine.

How long did it take you? Has it done as well as you thought it would be?

I think so. It's a pretty simple idea, at least in the first iteration, so it's kind of hard to mess up. It took me a weekend.

What is your business model plan for Gumroad?

Taking a fee for each transaction.

And you saw 10,000 shares only a day after launch. After the initial day 1 spike, how was growth the next few weeks?

It fell like crazy.

So do you think it can be a billion dollar business one day?

I think the idea can be. But only if I execute 100%.

Did you work on the project alone, or with others?

Just me. I consider myself a maker. I just want to build stuff. If that means designing, I'll design. If it means developing, I'll develop. They're both just a means to an end.

You've compared your project to Square, a mobile payments platform; can you explain that a little more?

Square makes selling stuff really easy. Gumroad does the same.

In addition to Gumroad, what other projects are you working on? How do you manage so many projects at once?

Gumroad is my only long-term project now. Everything else is a release-and-then-maintain project. So really I don't have to manage many projects at once, though it seems like it.

What advice would you give to someone about to build their own side project?

Stop reading this interview and start freaking building it. What are you waiting for?

DragonDrop



Mac App

Idea: Create a Mac app to improve drag and drop functionality.

Project Maker(s): Mark Christian

Download App: <http://itunes.apple.com/us/app/dragondrop/id499148234?mt=12>



Tell us a little about how you got started in web development and software.

I started programming when I was pretty young — in fact, I released my first piece of shareware when I was 14, back in the dark ages of the late 1990s. I'll tactfully leave out the name of that embarrassing first effort.

During my high school years, programming was my main hobby — I was constantly building little things. When it was time for university, I chose to pursue a computer science degree, and I've been working as a professional software developer since 2006.

What were you doing before you decided to build a side project, and how did it come about?

Some things never change: programming is still my main hobby, and when I have an idea, I'll often cobble together a prototype to I'd like to build myself. If you're really lucky when you do this, you'll discover a very interesting problem, and that's the beginning of a successful side project.

In the case of DragonDrop, my latest side project, there were two motivations. Firstly, I had an itch to scratch: I really like drag-and-drop as a way of moving things around on my Mac, but it really doesn't scale well to big monitors with dozens of windows open.

Secondly, my fiancée had an interesting project to keep her busy, but I did not. DragonDrop gave me something to focus on.

Why did you decide to launch this?

As DragonDrop developed, it became clear that it had the potential to be shared with the wider world. My fiancée, Nathalie Arch, was interested in helping me make that happen, so we decided to work together to finish it up and spread the word. Working with her on this project has been a wonderful experience.

When I started working on DragonDrop, I knew I wanted to build a simple app. There are other drag utilities, but I wanted one that felt like it could've been built right into the OS. This actually took the largest part of the development time — figuring out the user experience, and then polishing it to the best of my abilities. I knew I didn't want to leave a window floating on people's screens all day, but it took a while to think up the shake-to-activate gesture. Once I had that, though, the rest of functionality came together pretty quickly and I was able to devote myself to coming up with an app that was fun to use, and maybe even felt like it was built into the OS.

How did you get started?

One word: prototypes! I prototyped the basic "drag, drop, drag" functionality in Cocoa, and the shake-to-activate gesture using JavaScript. Ultimately, once I knew what I wanted to build, I threw away the prototype version and started fresh, just so I knew I'd be building on a solid foundation. I think a lot of side projects grow out of experiments, and that can make them hard to maintain and build on. Starting with a

good base is important.

If you started it while you were working, how did you find the time to do both, was it difficult?

I continue to work full-time at my day job, and plan on doing so going forward — at this point, I'm just not interested in "going indie". The key for me was to recognize how much slack time really existed in my daily routine: time commuting on public transportation gave me an hour and a half every day to work on little things. Additionally, I spent a lot more time idly browsing the web than I'm comfortable admitting. Once I was really fired up with DragonDrop, it was easy to fill the little moments in my day with useful work on it.

I also worked in my head during times when I couldn't work on my computer: figuring out next steps, trying to crack whatever problem I was facing in the code, imagining the web site. If you do this, then your time in front of a computer will be much more productive. For the same reason, keeping a simple to-do list proved invaluable. If I ever had 10 minutes to spare, I could always find something to do. All those little slices of time add up.

In terms of initial costs, I was able to leverage a few things I already had access to: I owned the domain name I wanted to use (shinyplasticbag.com, about \$10/year) and web site hosting (about \$10/month). Publishing apps in the App Store requires an Apple developer account (\$99/year) — something I already had from the release of my first app, Lidpop.

I also purchased some very helpful apps for my Mac: Acorn (for image editing), BBEdit (the best text editor on the planet), and DropDMG (for creating the installation disk image). All told, this means that publishing my two apps costs me about \$350 a year — less than a dollar a day. Not a bad investment.

What is your business model?

Developing an app gives me a very obvious business model: charge for it. I think the industry is at an interesting point in its history; people have gotten very used to free apps, but I think the tide is starting to turn back in the other direction. People are willing to throw a few bucks towards a developer who makes something they like. When I buy an app, I like to imagine myself treating the developer to a cup of coffee. If you use an app every day, isn't it at least worth buying someone a treat?

Even if you charge for your app, making money isn't easy, of course. If you're lucky enough to get some promotion, app releases and updates will cause your sales numbers to spike and then slowly fall back to a plateau. Development itself is punctuated with bursts of effort, followed by easier periods — consider those spikes to be your reward for the hard work you just did.

When you launched, how long was it until you saw your first sale, and where did it come from?

My side project had a bumpy introduction: after I submitted it to the App Store, it sat in limbo for a very long time. It was pretty frustrating to spend months working on something only to be stymied at launch by forces outside of my control. Ultimately, Apple rejected the app after about three weeks, claiming that the shake-to-activate gesture (which I felt was the most innovative part of the app) violated their App Store submission guidelines. I spent two more weeks trying to negotiate with them before giving up and making my own app store (see writing.markchristian.org/2012/03/30/make-your-own-app-store.html).

Even after I'd launched it independently, I kept trying to get Apple to relent. With the help of some very kind journalists, I asked my customers to contact Apple and ask them to change their mind — and it worked. About six weeks after my initial submission — two weeks after I started selling it independently — the app finally went live on the App Store. I'm not sure who the first customer was, but I thank every last one of them — they're all my supporters.

Can you walk us through the tools you used to build the initial version?

DragonDrop was mostly built with Xcode. I didn't preserve much in the way of early screenshots, so instead, I'd like to share a sneak preview of something that's coming up soon.



One of the biggest problems for DragonDrop has been getting new users to understand how to activate it. Lots of people get it instantly, but a large number of people have trouble performing the shake — they're too fast, or find it hard to do on a trackpad. DragonDrop doesn't currently offer any sort of visual tutorial, so I'm in the process of building a "practice" window to let people try it out and get feedback on what they're doing.

What parts of development do you do and what parts do you hire out?

With the exception of the beautiful icon, which was made by a pixel artist named Ben Rosaschi, Nathalie and I do all of the work ourselves. That includes the core creative work — the app, the web site, the screencast — as well as reaching out to potential reviewers and responding to customers.

Of course, we leveraged a lot of our social network to make everything happen. We roped friends into beta testing for us (thanks, Isaac, Adam, and Julie!) and made use of online programming resources for some tricky problems. No app is an island.

How are you marketing it?

Our main strategy has been to get in touch with bloggers who may be interested in the app and asking them if they're interested in a review copy. We've been very careful to select people who we believe will genuinely enjoy the app — if you don't do that, you're just a spammer. This strategy has paid off very well for us; DragonDrop would not be nearly as successful without the kind words (and feedback) of a handful of bloggers. Of course, we can't discount word-of-mouth advertising, either: we've been lucky to gain some very loyal followers who've told their friends and help spread the word.

What has been your biggest surprise so far?

Getting into the App Store after our initial rejection — followed by astonishment at the sheer volume of sales relative to the independent version. The App Store version easily outsells the independent version

40:1. It's a bit unfortunate, because Apple takes a substantial portion of each sale. Their 30% cut means that we end up clearing more than \$1 less per copy sold in the App Store. It ends up being worth it, of course — we have a smaller slice of a much larger pie.

What has been your biggest mistake?

Getting de-railed by the App Store review process. Waiting for Apple to make a decision took five long weeks, and took away a lot of my momentum. While I was waiting for an answer, I didn't actively work on polishing up the app. That was wasted time.

What tracking tools do you use?

We use Google Analytics on the web site, but it hasn't been particularly useful. The majority of our customers buy directly from the App Store, and never see the web site. Apple provides us with sales numbers every day, so we've been relying on that to see how we're doing.

Other than that, Twitter has been invaluable for keeping track of buzz and helping out customers who are having difficulties.

How is your project doing now, and what are your plans for the future?

DragonDrop has done better than we could've ever hoped. As I've said, I really just wanted to scratch my own itch, and then share my solution with the world. I build apps that make me love my Mac a bit more, and I hope they can make other people love their Macs, too.

I've got more app ideas in my head for the future, but the present is devoted to DragonDrop. There's still a lot more work to do.

What advice would you give to a developer that wants to work on their own side project?

Pick something that will challenge you, but not frustrate you, and make sure you're really into the idea. If your side project takes off, you're going to be spending a lot of time thinking about it. Make sure that excites you.

Unlock With Wifi



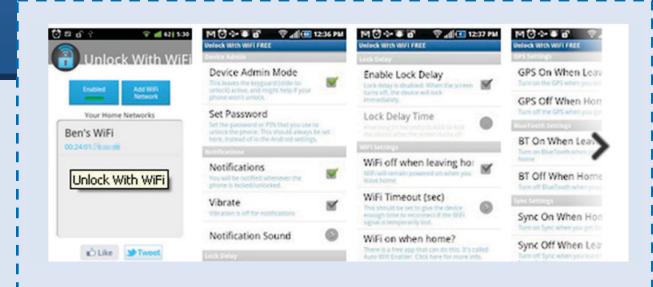
Android

Idea: Create a simple app to turn off wifi password requirement on Android phones.

Project Maker(s): Ben Hirashima

Download App:

<https://play.google.com/store/apps/details?id=com.benhirashima.unlockwithwifi&hl=en>



Tell us a little about how you got started in web development and software.

I got started in web development shortly after graduating from Santa Clara University, in Silicon Valley. It was 1996. I had majored in psychology, and didn't have any real job prospects. I was playing bass guitar in a band at the time, and our drummer had been a computer science major.

One day, after practice, he showed me Microsoft Frontpage, a WYSIWYG HTML editor. He said it was really easy to build web sites, and that people were charging a lot of money for them. I have always taken naturally to computers, but at the time, I had been focusing on other things, and barely knew my way around this new Windows 95 thing. I played with Frontpage, and learned to write some HTML by hand. I built a website for my parents' flower business, which they payed me for. By a stroke of luck, I ended up getting a temp job at Netscape through another band mate. The job was doing "discovery" (paperwork) for the impending Netscape vs Microsoft legal battle. In between reading Jim Barksdale's emails, and playing Quake on the company's LAN, I built an intranet website for the legal department.

Later that summer, I somehow landed a real web development job at a small web shop in San Francisco. My resume consisted solely of the work I did on my parents' website, and the Netscape legal department's intranet site. Companies were hiring anyone that knew anything about the web back then (1997), and the great thing about the web is that it's easy to show off the work you've done. It payed \$17 a hour, which was better than my temp job at Netscape, but was a joke in retrospect. Fortunately, I quickly moved on to bigger and better things.

Fast forward about ten years. I had gained a lot of experience in web development and had gone from HTML to more serious back-end development using .NET and databases. Smartphones began to hit the scene, and I was an early adopter. I began with a Palm Treo 600. I later picked up a Motorola Q, running Windows Mobile. Palm had a great threaded text messaging app in PalmOS, and I sorely missed that app on my Q, so I decided to build one for myself. I had been doing web programming in C#, and you could write Windows Mobile apps using C#, so it was easy enough to try my hand at it. I built the first threaded text messaging app for Windows Mobile, called TxtMan. It was a hit, and I had vague ideas about trying to make some money with it, but I just never got around to it. My web job was well paying, and kept me busy, so I gave away TxtMan for free.

A few more years went by, and I began to tire of web development. The web was no longer new, and the technology didn't excite me anymore. I had lost my passion for it. Work became monotonous. When Android debuted, I got a G1, and fell in love with it. My passion for technology was suddenly reignited, but in a field that I did not work in.

What were you doing before you decided to build a side project, and how did it come about?

In 2010, I decided that I was sick of my job, and quit after being with the same company for ten years. I had been saving money to buy a house, but decided that this was no longer a good idea for me. Instead, I used that money to take a grand trip around the world, and went traveling for six months. It was a huge step for me, and I was very nervous about it. I fretted that I was being irresponsible, and had no plan for obtaining employment when I returned. I managed to swallow my fears, and went on the trip that I had been dreaming of. It was an amazing experience, and I don't regret it one bit. 2010 will be a year that I never forget. I highly recommend that everybody do this at least once in their lives.

When I returned from my trip, I started playing around with Android. I wrote my first Android app, Bookmarks Light, followed shortly by Choose Browser. I distributed the apps for free, and they were well received. I liked Android, and I liked developing for Android. Android apps are mostly written in Java, and Java was an easy transition to make from C# because the syntax is similar. Of course, learning Java was not the only obstacle to creating mobile apps. There was much to learn about Android's APIs, and about mobile apps in general.

After making these two Android apps, I tried to get a job doing Android development. Just like my early days of web development, I had a couple of little projects under my belt to prove that I could do the work. However, things had changed since the early days of the web. Engineering interviews suddenly featured these ridiculous puzzle questions, and a battery of tests that resembled an SAT. I had no luck landing a mobile dev job, and was frustrated by the process. I wasn't going to study for these stupid interviews like a college student. I was an experienced professional!

Why did you decide to launch this?

Around this time, I had an idea for an app that I figured would really sell. The app would solve a problem that was bugging me personally, as a smartphone user. The problem was that I hated entering a password every time I turned on my phone. It's important to have a password on your smartphone to protect yourself in case of loss or theft, but it seemed unnecessary to be entering my password all the time while I was in the safety of my own home. I figured my phone should know when I'm at home, and not ask me for my password then. How would the phone know I'm at home? GPS? Maybe, but that would use a lot of battery, and GPS doesn't work well indoors. Aha! WiFi! And so Unlock With WiFi was born.

Unlock With WiFi turns off your password when you're connected to your home wifi network. You don't have to enter your password at home, and when you leave, the wifi disconnects, and the app automatically turns your password back on. It also features an optional lock delay. When you're not at home, the phone will stay unlocked for a certain amount of time after you turn the screen off. If you turn the screen back in within the time period that you set it to, it won't ask for the password. This is incredibly convenient if you're having a conversation over SMS.

Once you decided to do this, how did you get started?

I sat down and started making the app. Being unemployed, it was easy to devote time to.

What were your initial costs?

With Android there are virtually no startup costs. The development tools are free, and the developer account is free. The only thing you need is a computer and free time.

What is your business model?

I planned to go with a freemium model from the beginning. I have a paid version, and a free version of Unlock With WiFi. The free version only allows you to define one home network, while the full version allows unlimited networks, so you can also use it at work, for instance. The free version also has some options, like lock delay, disabled.

The conversion rate from free to paid is about 1%. This isn't unusual for a freemium model, but I feel it

could be better. I am exploring ways to improve this.

There is one aspect to my business model that is atypical. I was lucky enough to be approached by GetJar to participate in their GetJar Gold program. This is an invite-only program where GetJar gives away your paid app for free, and pays you a small sum for each download. This is their way of promoting their appstore. The amount they pay depends on the country your app is downloaded in. For the USA, Canada, and parts of Europe, they pay 20 cents per download. In other countries, they pay five cents. At first, I rejected the offer, thinking that GetJar would cannibalize my Android Market sales. However, after giving it some thought, I decided to try it, and I'm very glad I did. I now make more money from GetJar than I do from the Android Market, and it hasn't detectably reduced my sales on the Android Market.

I sell an average of 25 copies of the paid version per day on the Android Market, at \$3 each. Subtract Google's 30% cut, and that's about \$52 a day. I get around 350 downloads per day from GetJar at 20 cents each. That's \$70 per day. I currently only allow downloads in GetJar's countries that pay 20 cents, but I'm about to expand to all other countries where I'll get 5 cents per download. I expect revenue from GetJar to increase even more.

In the free version of UWW, I present the user with a notification after they've been using the app for a week. The notification says "Free upgrade available". When the user presses on the notification, I show them a screen that gives them a choice to upgrade to the full version for free from GetJar, or to buy it from the Android Market.

Unfortunately, I just today I received notice from GetJar that they are discontinuing the GetJar Gold program.

When you launched, how long was it until you saw your first sale?

I launched my beta on the xda-developers forum. I got a lot of testers, and good feedback since xda users tend to be more technically sophisticated. Launching on xda-developers gave me great exposure and built some buzz about the app. A few tech blogs like Lifehacker picked up the story from xda, and that drove a lot of people to download the beta, as well as the full version when it launched.

I have a Facebook page for the app, and I have Like and Tweet buttons in the app itself, so that people can easily recommend it to their friends. The Facebook page has 5200 likes, after a little less than a year.

I experimented with advertising on Google and Facebook, but found that the ads were not effective enough in driving purchases.

When I launched version 2.0 of UWW, I emailed a bunch of tech blogs about it, and offered them previews of the new version before it launched. A small portion of them replied, but the press was definitely helpful in driving downloads. There was a big spike in sales after the articles were published, but they returned to normal shortly after.

Can you walk us through the tools you used to build the initial version?

I used the Eclipse IDE, and wrote code in Java. It took me about three months to get the first version out.

What parts of development do you do and what parts do you hire out?

I do all the work myself.

How are you marketing it?

Sales of the paid version have been pretty flat. They spiked when I got some press about version 2.0, but then quickly returned to normal. I don't have the budget to do much advertising, and even if I did, advertising doesn't seem to be effective for paid mobile apps. I think most people find my app by searching in the Android Market, or through word of mouth.

Downloads of the free version, however, have been growing. Recently, I have seen an explosion of downloads in Mexico. The app has been translated into Spanish by volunteers, so that may have enabled this growth. The app has been available in Spanish for some time though, so I really don't know what has caused its popularity in Mexico.

In theory, downloads of the free version should help drive sales of the paid version. However, I haven't seen that happen, and I don't know why. Getting useful intelligence about why people are or aren't buying your app is difficult.

It has been almost a year, and the total number of downloads of the free version has exceeded 800,000. I'm getting over 8,000 downloads per day, and the app is moving up in the Android Market ranks. It's currently at number 68 in the Tools category. The paid version is number 190 in Tools.

What has been your biggest surprise so far?

Dealing with the public at large has been a challenge, and it's one that I wasn't mentally prepared for. When you have a lot of people using your app, there are bound to be some crazies included. As an app seller, you have to deal with some really stupid and unreasonable people. They're a small minority, but you still have to deal with them, and they do leave unfair and uninformed reviews on the Android Market. Eventually, my skin has thickened, but dealing with difficult customers is definitely my least favorite part of this job.

What has been your biggest mistake?

I'm glad I made Unlock With WiFi, but I have come to realize that it was a waste of time considering how much work I put into it. I have spent countless hours over the past year optimizing and upgrading the app. None of that work increased sales or benefited me in any material way. Sales have been decent for a utility app, but disappointing overall.

My biggest mistake was making an app that's in a poor selling category. If you look at data on which categories of apps sell the most, games are at the top, and tools/utilities at the bottom. My time would have been much better spent developing a game or other popular app type such as homescreen replacements, or music players.

What tracking tools do you use

I use Apsalar in the free version, but my paid version doesn't use it. I tried Flurry, but I prefer Apsalar's funnel analysis, which is the most important feature for measuring conversion rates.

How is your project doing now, and what are your plans for the future?

I'm working my butt off on a new app, that I have high hopes for. I started a small company with a friend, called Touch Pilot. We are making apps that allow you to control a game running on your PC from your tablet or phone. Our first product will be called Touch Pilot One.

Touch Pilot One allows you to control your flight simulator from your tablet. Adjust your throttle, mixture, prop, landing gear, flaps, and more with just a touch. Now you don't have to memorize all those keyboard commands! It works with any simulator or game running on a Windows PC.

<http://gettouchpilot.com>

What advice would you give to a developer that wants to work on their own side project?

If you want to make money, build an app that you know will sell, not something that's just a good idea.

Sticky Notes App



iPhone

Idea: Create sticky notes application for the iPod touch.

Project Maker(s): Mark Peterson

Download App: <http://itunes.apple.com/app/sticky-notes-pro-alarms-bump/id317468456?mt=8>

Tell us a little about how you got started in web development and software.

It pretty much started for me when the iPod Touch first came out, I heard the SDK would be dropping so I assumed it would be a great market to be a part of. With that, I had to teach myself how to program, with no prior knowledge.



What were you doing before you built notes app?

Before being a full time iOS Developer, I had a lead position in my local Airport. I maintained the cabin service crew.

Why did you decide to launch this?

Sticky Notes was originally a test project for myself, I wanted to see if I could do the basic concept of taking a screenshot of a note and applying it as your wallpaper. After a discussion with my partner, who is a graphic designer, we came up with tons more ideas and started making it into an actual app that we could launch.

How did you find the time to do both work/project, was it difficult?

At the time, as mentioned before I was employed at the local Airport. When a company bought ours out, they decided to do a re-hiring process. Fortunately (Yes, I say fortunately because if that hadn't happened who knows where I'll be today :)) enough I was not re-hired during this process. I began collecting unemployment, and started my journey of teaching myself how to program. Day in day out I sat at my computer reading all the documentation, blog posts, and example projects I can find.

What is your business model?

Initially, I didn't have any sort of business model. I doubt anybody coming from no experience in the market would have a business model. As time passed, the market grew and I adapted to it. You'll then mold yourself a business model.

So when you launched, how long was it until you saw your first sale?

Naturally, any developer submitting an App you'll get at least one sale on the first day. If you don't then there's a huge problem. There was a huge market for my Sticky Notes app. Apparently, the first month it netted ~\$15k in sales, on average ~\$500 a day @ \$0.99. It quickly rose to the top of the Utilities category ranked amongst the top 20. I contacted every review site, and blogger I could find to get them to do a review on the App.

Can you walk us through the tools you used to build the initial version of Sticky Notes?

The very first version of the Sticky notes App was developed on a hackintosh. It was a Dell Vostro 1500 that I converted to run OSX 10.5 at the time. It took me at least a week to get it working. After my first month in sales, I had enough to purchase my first mac, which was a MacBook Pro 13" Unibody.

That's impressive; you built a successful app without even having a Mac. How did you deal with technical issues?

Good ol' Trial and error :)

What parts of development do you do and what parts do you hire out?

I do all development myself; recently I've began doing backend work so I haven't had to do any hiring.

How are you getting users for the product?

Social Networks, contests, app Reviewers, and most importantly, have a product that people want and like to use.

How are you marketing it?

As of today, there are 3 different versions of the App. iPhone, Free iPhone, and iPad. All together we have nearly 4 million users.

After the initial release it rose to the top of its rank, naturally it begins to drop if you aren't pushing updates and getting it out for people to see.

What has been your biggest lesson so far?

Biggest lesson so far is being persistent. I've seen newcomers to the iOS Market drop like flies, if you keep charging at it you'll become successful one day.

What tracking tools do you use?

I use a number of tools to track the state of my Apps. I use MagicRank for Rank, AppFigures for sales, AppAnnie, and Flurry for Analytics.

How is your project doing now, and what are your plans for the future?

The App has seen better days; it's been running strong for 4 years now. We're soon launching a completely redesigned App and a Cloud service subscription. We're also in talks with a few investors showing interests in what we're doing. We'll see ;)

What advice would you give to a developer that is just about to begin a side project?

The 2 most important things:

- be persistent and don't give up easily.
- Partner up with a great graphic artist, preferably with iOS UI experience.

This is the combination to my success. :)

Hacker Node



iPhone

Idea: Create an iPhone app to browse the Hacker News website.

Project Maker(s): Sumeru Chatterjee

Download App: <http://itunes.apple.com/us/app/hackernode/id473882597?mt=8>

What were you doing before you decided to build a side project, and how did it come about?

Well I was working as an iOS developer in a corporation when I decided to venture out on my own and make awesome apps. One of the primary motivations for this bold move came from the success stories I had read on Hacker News and the knowledge I had gained about the technology business in general from HN. So to contribute back to the community I decided to make my first app a Hacker News reader app. It is called HackerNode and is available on the iTunes App Store.



What was your big differentiator?

The main thing that differentiates HackerNode from other hacker news reader apps is good UX. I have spent a lot of time to make sure the interface is smooth and intuitive. For example on the iPad you will notice that the cascading animations are very smooth and realistic. I have spent a lot of time getting this right.



HackerNode Artwork and UI

Once you decided to do this, how did you get started?

I already knew how to make iOS apps, so it was easy for me to start quickly and make a good product. Hacker News doesn't have a direct API, so I had to write a Hacker News HTML Parser in Objective C first.

If you started it while you were working, how did you find the time to do both, was it difficult?

I started after I had left my regular job.

What is your business model?

My business model is to sell apps for money and then spend less than I make!

When you launched, how long was it until you saw your first sale?

Initially this was a free app. But then after a while I decided to make it a paid app without any advertisements and got my first sale soon after.

Can you walk us through the tools you used to build the initial version?

Like other iOS apps, HackerNode is built with XCode and written purely in Objective C. I have used the 320 framework extensively for building this app.

It took me almost a month to get the MVP ready.

What parts of development do you do and what parts do you hire out?

I have done everything for this app.

How are you marketing it?

When I launched it on Hacker News initially I priced it at \$2.99. The number of sales for the first three days after launch were 289, 89 and 52.

Currently HackerNode gets about 4-6 downloads per day on average. I don't really market it much but I'm working on that department. However I do some minimal marketing on Twitter that I think is a major source of sales.

What has been your biggest surprise so far?

One of my surprises has been the fact that the number of downloads in the US is much greater than all the other countries combined.

What has been your biggest mistake?

I think I overestimated the size of the Hacker News iOS app market. But it has been a good learning experience so far.

What tracking tools do you use?

I use Google Analytics for all my tracking, website and mobile.

How is your project doing now, and what are your plans for the future?

The project is giving me a decent side income of about \$300 per month now. I do intend to make improvements and add cool new features such as recommendations, archives etc.

What advice would you give to a developer that is wanting to work on their own side project?

I think that a lot of fellow developers like me think too much before making their first product. My opinion is that it is pretty much a waste of time to think too much about business models when you don't have any experience selling software or marketing it. So I'd say as a first project just makes something quickly and start learning these things from real world experience.

DotNetInvoice



Browser

Idea: Acquire an app and turn it into a consistent passive income.
Project Maker(s): Rob Walling
Website: <http://www.dotnetinvoice.com/>

Tell us a little about how you got started in startups

I started launching web apps in 1999 when I was a year out of college. I grew up in the bay area and I loved startup culture, though I didn't have a clue what it took to actually make one successful. I launched 5 or 6 failed attempts over the next 7 years before my first real success.

What were you doing before you acquired DotNetInvoice?

I was working as a freelance developer. I enjoyed the freedom of location and the ability to work hourly was great – when my wife went out of town I would work long hours and make great money. But I didn't like the feeling that I was running on a hamster wheel from one project to the next. I constantly asked myself if I was going to be running the same race when I was 50.

Why did you decide to acquire this?

I saw a forum post from one of the previous owners who mentioned they were looking for a partner to help them market the app. Starting a partnership with someone I had never met is not my idea of a good decision so I asked them if they would consider selling it. We worked out a deal over the next few days and I owned it by the following week.

I had only acquired one app before this, but given how much time I spent consulting and the hourly rate I was able to charge, I essentially had more money than time. Building DotNetInvoice from scratch would have taken 300-400 hours and I got it for a fraction of the price I would have spent had I taken time off work to build it, or hired it out.

If you acquired DotNetInvoice while you were working, how did you find the time to do both?

It was difficult at first. I worked evenings and weekend cleaning it up, fixing bugs, revamping the website and learning marketing.

What is your business model?

DotNetInvoice is web based invoicing software written in ASP.NET. Customers purchase it for a one-time fee of \$329, download it and run it on their own server. Source code is included so they can customize it as they wish.

So when you launched, how long was it until you saw your first sale?

After I acquired it the first sale came in within a week. It sold 3 copies during the first month I owned it, but I had expected it would sell 8 or 9 copies based on discussions with the previous owners. I was really disappointed and figured I had made a huge mistake.

What tools and languages did you use to build it?

It's built in ASP.NET on SQL Server, so I use Visual Studio.

What parts of development do you do and what parts do you hire out.

I have a business partner who handles development these days while I handle marketing. I have other

apps that I own where I still write code, but I haven't touched the DotNetInvoice code base in a year or two.

How was your growth during the first three days, three weeks of launching?

After the initial month of poor sales I talked with a few trusted advisors and tripled the price from \$98 to \$295. I also worked 60 hours during nights and weekends in about 2 weeks to fix more than 20 showstopper bugs. I released a new version and sold 3 copies the following month, for a 300% revenue increase. As I focused on marketing over the next few months I was able to double it again.

What has been your biggest lesson so far?

Whether you build or acquire, you are going to face major setbacks along the way. These are problems that you don't think you'll ever overcome. The best entrepreneurs I know decide that they will not be beaten by these kinds of problems. Everyone else calls it quits.

Has it done as well as you thought it would?

I have been impressed with the consistency and longevity of DotNetInvoice, but I have to admit that I had hopes for the first few years that I would be able to grow it larger than I was able to. We tried expanding the market from the vertical niche where it sells best, but were never able to substantially grow it.

What tracking tools do you use?

I use Google Analytics, KISSmetrics and HitTail (disclosure: I own HitTail).

How is your project doing now, and what are your plans for the future?

I've owned it for over 7 years now and it's been a consistent earner, but has been very difficult to grow due to the rise of SaaS invoicing apps, and the fact that it's not something I focus on full-time. But after owning it for this long, and the fact that it was what I consider my first success, I plan to keep it for the foreseeable future.

What advice would you give to someone about to build their own side project?

Building a business is a combination of long hours and intense mental challenges. Most people quit way too early. If you aren't prepared to put in 400-600 hours of marketing and coding before your launch day, and at least that much in the 6 months following, don't try to build and launch a side project.

I would also highly recommend finding some kind of accountability partner(s); someone who can keep you on track and motivated during this long process. Without someone else who's going through a similar journey, it's a long, lonely road.