77

Failed Startup Post-Mortems

Discover the reasons behind startup failure, from the founders themselves.

77 Failed Startup Post-Mortems

77 assorted essays and post-mortems from the founders or investors of failed startups, collected from around the web.

All stories are attributed with their original source.

This book is not intended to be republished for any kind of financial gain or sold for profit, all articles remain the property of their respective writers/owners/sites.

I have merely packaged this information for my own (and your) reading convenience, so we might read through these amazing stories at our leisure on our Kindles, phone or ipads or whatever.

I'm not profiting, looking to profit from these collected works. Just packaging them together so others may enjoy them freely, as I have done.

I've organized essays purely by the startup that failed, for ease of use. Also in my mind was the need to preserve some of these startup stories that we can all likely learn a lot from. Some have already started to disappear as domains expire, so I've gone to the trouble of getting those articles too, via the wayback machine.

Note that images contained in the original articles online have been left out mostly, in the interests of saving time and file size.

Putting all of these essays and posts into one ebook has taken me a lot of hours already. There may be typo's and poor grammar, but don't shoot the messenger - I've just collected content that was already out there, I have NOT edited or proof-read everyones articles for them.

I hope you enjoy the convenience of reading all of these amazing articles in one easily digested ebook, as I have.

There is much within these pages that can be learned and applied to our own startup projects.

This books 'curator' (on twitter): @edward_b

Startup Failure: 99 Dresses

My startup failed, and this is what it feels like…

Over 90% of tech startups fail, but I never thought my baby, 99dresses, would be one of them.

If there is one thing that doing a startup has taught me, it's that I am much more resilient than I could have ever imagined. Looking back, when I started 99dresses fresh out of high school I was very naive and had zero idea what I was doing. In fact, I didn't even know what a startup was! I just knew I wanted to solve a problem I personally experienced: having a closet full of clothes but still nothing to wear.

Since then I've survived being stabbed in the back by cofounders, investment rounds falling through, massive technology fuckups that brought sales to a halt, visa problems, lack of money, lack of traction, lack of a team, hiring the wrong people, firing people I didn't want to fire, lack of product-market fit, and everything else in between.

And yet I failed. I won many battles but I lost the war.

I take complete responsibility for this failure. Were other people involved in 99dresses? Of course. Was any of this their fault? Absolutely not.

The startup press glorify hardship. They glorify the Airbnbs who sold breakfast cereal to survive, and then turned their idea into a multi-billion dollar business. You rarely hear the raw stories of startups that persevered but ultimately failed — the emotional roller coaster of the founders, and why their startups didn't work out.

As things were looking bleak at 99dresses I started seeking out these stories, desperately hoping for someone — anyone — to relate to. Failing is lonely and isolating. Every time I'd scroll through my Facebook feed all my startup friends were launching new products on Techcrunch, announcing their new fundraising rounds or acquisition, and posting photos of their happy teams. Ask any founder how they're doing and you'll hear something positive. Whether that's the truth or not, that's what we're trained to say.

I found postmortems of startups outlining what didn't work and why the company went under, but I was hard pressed to find anything that talked about the emotional side of failure — how it actually feels to invest many years of your life and your blood, sweat and tears, only for your startup to fall head first off a cliff. Maybe it's because most founders are men, and men generally don't like talking about their feelings. Maybe it's because failure is embarrassing.

I don't know why this is the case, but here is my contribution to the cause: my story. This is what failure feels like. I hope it helps.

Where it all began…

Many startup folk say that failure should be celebrated. "Fail fast, fail early, fail often!" they all chant, trying to put a positive spin on the most excruciating pain any founder could experience.

Let me tell you — failure fucking sucks. If I would have failed fast, early and often then I would have given up 99dresses years ago when, in 2011, I travelled to my parent's place in the countryside of Australia, locked myself away in my room and cried for what seemed like an entire week. I had launched 99dresses in Australia 9 months earlier and received some great traction, but I was losing momentum due to technology problems that I didn't understand and battling a whole host of other issues.

I felt like I was drowning in a black ocean, and I couldn't see any light at the surface. I didn't know which way to swim.

At the same time the Australian press would continue to approach me for interviews. The fact that I was a teenage girl working on a startup in a male dominated industry seemed to garner a lot of attention, and I'd take the interviews that came my way because that was my job. It was my job to be positive and paint a happy picture for the media, who seemed to talk about me as if I was some kind of entrepreneurial wunderkind because of my age and the fact that I had breasts.

This didn't help my impostor syndrome — the constant feeling that everybody was always giving me way too much credit. I

remember one reporter saying "you must be so proud of what you have achieved" and I was completely stumped by that statement because I'd never actually thought about it. Was I proud? What had I actually achieved? We had some traction, sure, but we also had many problems that needed solving. I was just waiting for the day when everyone would figure out that I'm not that extraordinary.

"But you're taking a massive risk! That's so brave!" they'd say. I never thought so. The biggest risk in my eyes was going to university, getting a stable job, and sliding into a comfortable life. There's nothing wrong with that, but I knew it wasn't me.

Plus, the worst that could happen if I failed was that I'd end up living with my parents. I think the really brave founders are the ones who will be out on the street if they fuck it up, and still do it anyway. It's easy to take risks if you have nothing to lose.

My mother said "Nikki, are you sure that you really want to do this? It is so much pressure for a 19 year old to take on. No one will think less of you if you decide this isn't what you want". My parents are my number one supporters but my mum hated seeing me in so much pain, even if it was character building.

But despite the horrible sinking feeling in my stomach, and the fact that I had no money left, and the fact that I had no stable team, and massive product problems, and was feeling burnt out, and had no idea how to overcome any of the aforementioned obstacles, and felt completely alone in it all, I persevered.

I didn't fail then. I couldn't fail. This was my baby, and if it was going to fail it would be over my dead body.

I became numb to the pain, and despite waking up for weeks on end with no glimmer of hope and no desire to get out of bed, I still made myself sit at my desk and work.

Eventually, things took a turn for the better.

When you're at your lowest, the only way forward is up

I applied for a university team business planning competition with a $10k prize, paid a friend $500 of the prize money to be on my 'team' so I could qualify to enter, wrote a winning business plan and took out first place. That was enough money to buy me a plane ticket and some accommodation to the US.

I met my friend and advisor, Matt, who took me under his wing and helped me more than I could ever have hoped. My developer was admitted to hospital with a very serious illness and dropped out of the company, but I replaced him with 2 co-founders. I got into Y Combinator and headed to Silicon Valley—startup Mecca for a starry eyed young founder like I was—for 5 months. We rebuilt the 99dresses product and launched it in the US. We were getting traction. I signed a $1.2 million seed round with a group of investors on a valuation cap that I honestly thought was ridiculously high.

99dresses was back, baby!

And then, all of a sudden, we weren't.

Another trip down the emotional rollercoaster

I had to fly back to Australia to get a working visa as soon as the funding paperwork was signed, and the next day my two "co-founders" decided to tell me they were leaving the company without even a hint of warning.

The $1.2 million hadn't hit our account yet, but even if it had I would have felt uncomfortable accepting it with no team in place to execute my vision. I would have looked like a fraud and an idiot anyway—what kind of founder announces to her investors that she suddenly has no team the day after she takes their money? And furthermore, how could I not have seen this coming? I was completely blindsided.

I went over to Matt's office, and he proceeded to pour vodka down my throat whilst telling me I was much better off without them. Like most of Matt's lessons it was hard to see that then, but he was right.

The next day I rang up our lead investor who decided to pull out of the round. Then another investor fell off. Everything I

worked so hard for was crumbling to pieces. If only I'd closed everyone individually, instead of agreeing to round up at least $1mil to get the lead on board. But then I realized that these "co-founders" would have left anyway, leaving me in this same position.

I was stuck back in Australia still with a big vision, but as a single, non-tech founder with no team, no product (I needed these co-founders to keep the product running), no US visa and just some money that I'd gotten from being a YC company.

I remember my sister taking me for a walk after it all happened. She sat me down in a park overlooking Sydney harbor at night time and made me listen to 'Shake it out' by Florence and the Machine. She told me I'd bounce back, that I'd overcome this like I always did. I wasn't sure I believed her, but I knew I'd survived worse. This ended up becoming my motivational song that I would listen to when times were tough, because it reminded me that I could surmount huge obstacles if I wanted to.

I didn't fail then. I just started again.

Starting over

There were 5 investors who invested in me, despite all of this. They believed in me when I was having trouble believing in myself, but I couldn't show them that—that's the cardinal sin of any entrepreneur. Always be confident. Always be smiling. Always stay positive. Sell, sell, sell!

I remember one investor sending me an email saying "Shit happens. Take the money and go sort it out." Another told me to go make him some crack for women.

My cap got sliced in half, but at least I wasn't broke again.

So I closed $595k and started looking for a new co-founder. Problem was, I didn't trust anyone. Not after what my previous co-founders had just put me through.

But then I met Marcin, who quit his corporate IT job and joined me in an office we referred to as 'The Cave' because it was cheap and nasty and had no natural light. I remember he came in on his first day, and midway through a conversation my chair completely collapsed. The next day he brought in his own chair. I was very jealous.

We rebuilt 99dresses again and launched it in the US which was proving to be ridiculously hard when we weren't physically in the US and having to handle some stock and seed a community from another continent. We were having trouble getting traction. The market had moved on, competitors had flooded the space and the product we had built just didn't provide enough value in comparison.

Add to that the fact that we were building a 2 sided marketplace, and you might get a sense for how tough things were. The US market is huge, hyper-competitive and way harder to crack than the Australian one. We were frustrated by our lack of progress, and the product I'd promised our investors just wasn't working.

I didn't fail then. We pivoted.

Our big pivot

I caught a plane to the US and talked to as many women in our target market as I could. We interviewed more customers. We discovered a very clear set of problems that explained why our product just wasn't working in the US market.

I rang up the team in Australia, and told them, quite bluntly, that we needed to chuck everything out and approach the problem from a different perspective. I presented a new idea for a product that seemed to resonate with the girls I was talking to. The team did not take it well, and I definitely communicated the change very poorly. I almost got on an early flight home because I felt a mutiny brewing—we were throwing out many months of hard work. This wasn't my finest moment as a leader.

Despite this, the team rallied together. We threw out our website and concentrated entirely on mobile. We had a mobile website prototype in front of users within a week and iterated based on that before building out the native version.

We hustled to get anyone we could to try out our beta app. We must have emailed thousands of bloggers, and some ended up giving it a go. Items were being traded, and girls were paying us money. This new thing was working! We couldn't wait to launch it in the US, but we needed to physically move there first in order to do things properly.

Visa issues

Problem was, we didn't have any visas. You see, it's very easy to get into the US as an Australian if you have a degree in a specialized field, which I did not. Marcin had to wait it out to first become an Australian citizen with his wife, then get his E3 visa. However, right before joining 99dresses his wife had fallen pregnant with their first child, which they needed to give birth to in Australia 9 months later. Marcin was then tasked with moving his wife and baby halfway across the world to chase our startup dreams. Needless to say, he's a very brave man.

I, on the other hand, was faced with my next big challenge: proving that I was 'an alien of extraordinary ability' that was worthy of living and working in the US without a degree (after all, I gave up my scholarship and dropped out of university when I got into Y Combinator). After about 7 months of working on my petition, I was ecstatic and incredibly grateful when I got approved for an O1 visa.

I practically skipped over to the US consulate in Sydney for my appointment, where I was to pick up the visa. Instead, I was interviewed by a lady who took an obvious immediate disliking to me. She told me she was putting me through extra processing, so I wouldn't be getting my visa that day. She told me it was random. She told me it would take 2 weeks.

I later found out this processing was not random—it was reserved for potential terrorists, and could take up to several years.

As an entrepreneur I HATE feeling helpless. I'm used to taking action on something and producing some kind of result. I like being in control. In this instance I felt completely helpless, and my startup was at the mercy of a government worker on a power trip.

We were already running behind on launching this app in the US, and the consulate had my passport. I couldn't get out of Australia. The consulate made me jump through hoop after hoop, and a few months later I still didn't have my visa. It got to the point where I had to call the consulate hotline every single day and split test different types of crying (machine-gun bursts of sobs vs. long sad silences vs. loud ugly cries) on the operators (males were much more receptive to helping out), and occasionally I'd get lucky and have one of them put in a report for me. I hated doing it, but it was the only way to push things forward.

I finally got my visa, and took the next flight I could get out of Australia with four suitcases—2 full of clothes, 1 full of shoes and another with all my electronics and miscellaneous items. The contents of these suitcases just about summed up my life.

I'd achieved my dream of moving to NYC, and I was living in a shoebox. It was all I could afford on my startup salary.

Soon after, my 25 year old sister and 19 year old brother both bought gorgeous apartments in Sydney. Whilst I was absolutely thrilled for them, I also couldn't help feeling a little jealous as I sat in my tiny convertible bedroom with no windows. If this all didn't work out I'd be financially left with nothing, whilst my siblings were off investing in their financial future. That didn't really scare me—I've realized that money isn't a huge motivator for me—but it did flare my competitive side. We probably all compare ourselves to others way more than we should…

Re-launch time!

After hiring a few people and finding an office in NYC we were ready to launch. We solved the chicken-and-egg problem using techniques that we promised never to speak of again because they squarely sat on the grey/black spectrum of naughtiness. If there was a line, we definitely crossed it. We had to. These hacks were harmless to others, so I figured it was only a problem if we got caught.

Our plan worked better and faster than I'd budgeted. Within 3 months we were doing over 1,000 trades a week, and bringing in revenue on every trade. We continued to grow.

Our app store reviews were overwhelmingly positive. Obsession did not begin to describe how some girls treated

99dresses. Within a few short months several power users had spent over $1000 each and traded hundreds of individual items. We steadily grew our stock turnover rate from 17% to 50%—that was 2-3x better than our competitors. Everyday I'd be wearing a new outfit that I'd received off the app. Our retention rates were really exciting. If my investors had wanted crack for women, then that is what we had created.

Based on the way we were growing, we thought we could get cash-flow positive before our funding ran out.

I had 99 problems and our runway was one… But then growth started to slow down. The average value of items listed steadily declined and our fees were based on this value, so although we were growing transaction volume our revenue wasn't budging. We started to see some holes in the business model. Whilst our retention was great, we worried about our activation rate.

In an attempt to save ourselves we made one more pivot; this would turn out to be our last one. The pivot made complete logical sense based on all of our research, but introducing it to our community was a nightmare. There was mutiny within the app. While our top line metrics shot up in a massive way, our one metric that mattered—transactions—plummeted.

Meanwhile, I had approached our existing investors about getting a bridge. I knew we had something really special with amazing potential, if we just had enough runway to give it an extra push. I also knew we weren't perfectly poised to raise a bridge round, unless our existing investors were going to pony up the cash. We'd been in the market a while, and although we had to overcome a number of setbacks to get out here, that didn't seem to matter too much to external investors. Bridge rounds just aren't that sexy.

We only had one institutional investor in our previous funding round, and I was so relieved when they told me they wanted to lead this bridge. Boom! It looked like we were going to live to see another day.

I sent through the due diligence documents and worked with them to answer all their questions.

They were taking longer than anticipated to get back to me so we could get the deal done and move on. Then one Wednesday I got a call from them, and the line was kind of crackly. However, it sounded like they not only wanted to lead, but they actually wanted to fill up the entire round! Relief flooded through my body. I was so nervous.

Then I heard a 'but…'

And the rest of the conversation explained why they would not be doing that. My stomach dropped. I knew they were our best shot of getting the money, and some of the angels who had previously invested were interested in coming in but only if I could get a VC to lead it, probably for some oversight. We now had very little cash left, and very little time to find someone else.

Turns out, under closer scrutiny some of the other partners in the firm didn't like how competitive the market was. 99dresses was squarely focused on trading cheaper fast fashion (fast fashion is really hard to re-sell for cash), but all the competition were mainly focussed on buying & selling designer fashion. Despite our differentiation, the space is crowded and the competitors are well funded to the tune of tens of millions of dollars each.

I felt my voice crack whilst I was talking back on the phone. I was trying so hard to hold it together and be professional, but I could barely speak without it being obvious I was crying. Damn emotions! I was embarrassed.

Our last attempt

It was night time and I walked over to Marcin's home in tears, fully expecting him to take the safe and responsible route of deciding to get another job. He had a family to support, and I felt an extraordinary amount of guilt for putting him in that position.

Instead, Marcin surprised me. He wasn't willing to give up that easily. None of the team were. I was taking on this massive burden and internalizing everything, when in actual fact my team was prepared to fight to the end alongside me. We made a plan for cutting our costs to extend our runway whilst we tried to get some more cash in the door.

The next day I gave notice on our office, and let someone go. We were already a very lean operation, but now the work of 2 was being done by 1 person on operations, and we shifted our focus to only the most essential tasks to buy us more

time.

I didn't tell many people about what was happening. You're not supposed to talk about this shit. If someone asks how your startup is doing, you fire off some kind of positive phrase like a reflex. My friend gave me a hug and told me to go read 'The Hard Thing About Hard Things' by Ben Horowitz. I bought the book and sat in a coffee shop that Saturday afternoon reading it through. I identified so much with the struggle—I'd been through it many times before whilst aboard this emotional rollercoaster.

I realized something: I was fucking tired—physically and emotionally. I wasn't sleeping properly. I hadn't been on a proper holiday since our 'schoolies' beach celebration straight after I finished high school in 2009. The holidays I had tried to go on just ended up being long strategy sessions in my head to figure out my next move whilst lying beside a pool. All I could think about was this damn startup and it was completely consuming me.

I had no bandwidth for anything else. When someone asked what hobbies I had outside of work, I'd laugh. I'd recently started having mini panic attacks whilst I was doing ordinary things, like taking a shower or doing my hair. I felt like a shitty friend. I couldn't even contemplate having a relationship (I tried that before, but yet again this startup won out over him). I wasn't sure how much longer I could do this.

My mother told me to trust my gut. If my gut told me that I didn't have faith in the business, then there is no shame in winding down the company and moving onto something more productive instead of raising more money. I'd learned an awful lot in the past few years.

I told my mum I didn't trust my gut when it came to this. My gut was telling me to quit. Problem was, my gut had told me that before in my darkest hours and I still pulled through. If I had trusted my gut then I would have quit years ago.

I knew the only way this was going to die was if we were killed. I am not a quitter. I owed it to myself, my team, my investors and the 99dresses community to see this through.

I continued approaching investors without luck. I'd be invited to cocktail parties full of VCs where I'd don my painful sky-high heels because I'd split tested heels vs. flats, and for some reason a 5'11 woman in 7 inch heels commands more talking time and attention from investors than one in the comfy flat booties I wear to work. Apparently height gives you presence. Once or twice I'd have an investor asking if I knew what an angel was, or if I also modelled because of my height, or some other unintentionally patronizing comment that I doubt any guy would be subjected to. I learned to take it all in my high-heeled stride.

I kept hearing the same thing from these investors. "That's a very interesting business, but we'll either put in the first money or a series A. We don't do in between. I'd love to keep in touch though, and see you progress to a series A where we might be able to help. Oh, and why aren't you getting this bridge from your current investors?"

I remember one day Marcin joked that I was a control freak, and I was really surprised. I'd never perceived myself that way—I just liked things done a certain way and to a certain standard that matched the vision in my head. When it came to non-99dresses related stuff, I thought I was pretty chill.

Over the past few weeks leading up to this event I did start to get a sense for what he was talking about, though. I wasn't a control freak in that I was obsessed with controlling outcomes—I was a control freak who just needed to be in control of the inputs.

This became more obvious as everything started looking more and more hopeless at work. I started eating much healthier, strictly cutting out wheat, sugar and anything processed. To take a mental break I would read about bio-hacking, which is incidentally all about understanding and controlling how inputs effect your body. I told myself this would give me more energy to hustle, but really I think I just had to feel like I had control over something—anything—when my startup's fate felt so out of my control.

Closing down

With a few weeks of cash left, Marcin and I agreed to use our remaining time to shut down the app gracefully for the sake of ourselves and the community. I came into the office that day prepared to have a hard conversation with him, but we both looked at each other and knew it was over. There were some tears, and I was grateful to have a curtain of long dark

hair to hide my bloodshot eyes behind as I walked through our co-working space. I felt physically sick all day, and my stomach wouldn't let me keep any food down. I lost my appetite for the rest of the week.

My first instinct was to apologize — to Marcin, to my team, to my investors, to the loyal community we'd built. I felt shame, guilt, embarrassment — like a shepherd who'd led her sheep off a cliff when it was my responsibility to keep them safe. I logically knew that I shouldn't feel these things, but emotions aren't always logical.

In fact, I didn't really know what I should be feeling. I'd been working on this company ever since I finished high school, so 99dresses was all I'd ever known. It was a huge part of my identity — I was "that 99dresses girl". Who was I without this startup? I had no idea. Just an ordinary girl, I guess.

My friends invited me out to drink away my sorrows and get my mind off things, but I just didn't feel like it. I was scared I'd meet someone new and they'd ask me what I do, and I wouldn't know how to answer. I was also embarrassed because I couldn't afford to pay for anything superfluous anymore — I still don't know how I'm going to pay rent at the end of the month. As a woman going out in NYC my nights were normally cheap because cute guys would buy me drinks, but I am not the kind of woman who expects that. I'm independent. If I couldn't pay for myself then I wasn't going out at all.

I wasn't depressed so much as disappointed. I tried so fucking hard, and I still couldn't make it work. There are many things I would have done differently were I to do this all again, but Marcin and I agreed not to get sucked into the 'shoulda woulda coulda' trap. "No regrets", he said. We both learned some hard lessons from our mistakes, but it also made me realize how much luck and timing are often huge factors in success and failure.

The next day a report came out by a startup with a very similar model to us, but in a different vertical. We'd traded 3x more items than them in our first 8 months of the US app being live, had 2.5x more members and had a business model in place — all with a team half their size. They'd gone on to raise a sizable series A; we'd failed. Our investors said we did a lot with the money we had.

It's easier to accept defeat if you try and try and try but don't get anywhere. You call it a failed experiment. The failure is easy to justify.

It's incredibly frustrating to try and try and try, and when you finally start to get some good traction you fall off a cliff. Our business still had problems, sure, but so does every other startup.

Moving on

So this is where the story comes to a close. My friends all ask me if I'm fine, and I honestly think that I am. It's been a wild ride, but it's time to move on.

A cruel consequence of my failure is losing the US visa I worked so hard to obtain. Once I stop being the CEO of 99dresses I technically have 10 days to sell all my possessions, pack my bags, say goodbye to my amazing team, my friends and the life I've been building here, and leave.

That being said, I'm excited to start a new chapter. As much as I love startups, it's somewhat liberating to have no responsibilities to anyone but myself — no team, no investors, no customers to look after. Maybe now I can be a normal 22 year old for a while: indulge my wanderlust, make some bad decisions, try something new.

I'll be taking some time out to recharge whilst living with my parents in a country town of 2,000 people where the internet is slow and there is no Seamless. I hope I survive. Honestly, I'll probably get bored within a week and start working on a new idea. I already have a few.

When I started 99dresses I was going to go big or go home. It's been a great adventure, but now I'm going home.

The end

So that's it. That's my story of what failure feels like. I hope reading it was as helpful to you as writing it was cathartic to me.

Most startups fail, and yet this industry doesn't talk about failure nearly enough. I'd encourage anyone who has failed to

write about how it felt, as I can't tell you how much that would have helped me in those final months & weeks. I just wanted someone to relate to. Instead, I was left feeling isolated and ashamed.

In fact, I thought it might be therapeutic to curate a collection of stories from founders who have failed and put them together in a book. It might be a little project for me whilst I take some time off, and I'm sure it would be helpful to someone in my current position.

If you want to get involved or contribute your story then shoot me an email. My email address is nikki @ 99dresses dot com (yep, I'm going to need a new email too—still haven't sorted that out yet).

Thank You

I thought I'd end this post by publicly thanking everyone who has been a part of the 99dresses journey.

To my co-founder, Marcin—there is no one I would have rather had by my side through this experience. The sacrifices you made to make this all happen are nothing short of inspiring. If Zoe was old enough to say anything other than "this!", I'm sure she'd tell you how proud she is of her dad. I'm going to miss your terrible jokes.

To the team, past and present—thank you so much for all of your hard work, perseverance and loyalty. Chandra and Oguz, you guys are amazing. I loved coming in to work every day because you always made it fun. I'm going to miss you all immensely. When I do another startup I'll be coming after you guys again :p

To the wives, Natalie & Semiha—I'm sure it's not easy having your husband involved in a startup. You were both so supportive of the long hours, sacrifices and emotional ups and downs, so thank you.

To my family—you guys have always been my #1 supporters. A special thanks to my mum for putting up with me, even when I was a stressed out and horrible daughter. You're the strongest woman I know, and I hope one day to be as bad-ass as you.

To my friends—I couldn't have done this without your support. You celebrated my highs and comforted me in my lows, and for that I'll always be grateful.

To Matt—You're the reason I got this far in the first place. If it weren't for you I probably would have failed in 2011.

To my investors—thank you for believing in my vision and trusting me to fulfill it, even through the rough patches. I really do appreciate it.

To our community—without you, we'd be nothing. Thank you for loving 99dresses, and spreading the word. Many of you I now consider friends, and I'm so grateful for your support and loyalty.

Ok, that's it. It's over. Now it's time for a nice long sleep.

Original article by Nikki Durkin here: https://medium.com/@nikkidurkin99/my-startup-failed-and-this-is-what-it-feels-like-c5d64b3ae96b

Startup Failure: Dinnr

Seven lessons I learned from the failure of my first startup, Dinnr

(Background: Dinnr was an ad-hoc, same day ingredient delivery service. Select a recipe on our website, and we deliver everything you need to cook that recipe at home, all the items pre-measured with printed instructions. All you need at home is oil, salt and pepper and a reasonably equipped kitchen.)

It all ended on 12 January 2014, when I decided not to accept a small second round of investment I had secured via crowd-funding platform Seedrs. Seedrs knew about the difficulties we were in and we agreed with the Seedrs team that it was best to cancel the investment. I had run out of ideas on how to turn things around, how to make people buy our product. The only thing I would do with this money would be to try to acquire more infrequent customers and keep the lights on for another few months. No way it would be an investment that would give investors a 10x return.

It wouldn't have been enough money to pivot to another business model. No one (myself included) even had a good hypothesis on what it could be that would work in London and that we could leverage our assets for.

I wasn't particularly tired. Even after months of fruitless trying, I was as keen to make things happen as in the early days, but I did feel like I arrived at the end of a dead-end street. Pulling the plug radically was the only reasonable (and ethical) thing to do.

In those days, I spent 9 hours a day, 6 days a week, sitting in a cold warehouse in West Acton, hot air blowers creating 25 degree bubbles around parts of my body in an otherwise 14–15 degree room. The large industrial fridges were empty. Long gone were the days I pre-stocked ingredients for faster order fulfillment. Now, when an order would come through, I would jump on my bike and buy fresh ingredients at a nearby superstore. After 15 months of operations, we had approximately one order per day, generating on average £26 in revenue and supposedly 30% in gross margin, but given the low volumes, we were losing money on every order.

Now, with the distance of a few months, it sounds positively crazy to run a "business" like this. But of course, one sticks to the guns because one has set out to make things happen and one has an obligation to investors. And then there's the adage of "a winner never quits", tales of entrepreneur legends who lost all their money, got abandoned by family and friends, were so hungry that they ate cardboard, only to triumph in the end and build a global empire.

But I had realized that it was over. Dinnr simply didn't have legs (never had them to begin with, to be explained below) and I had reached an inner state where continuing would have been insane. After having spent tens of thousands of pounds of my personal money and working obsessively on an idea that had inspired me for a long time, I felt that I had given it all and that it was time to move on.

This rather long blog post is about my lessons learned and is a post mortem. I will try not to go too much into the Dinnr specifics. Most of the readers of this post will not be interested in Dinnr itself. Instead, I will try to reflect on my 18 months and 12 days of entrepreneurship and see what I can come up with that will be applicable to any startup.

Rock'n'Roll.

Lesson #1: Are you solving an actual problem?

I will never forget the shock of the first week, September 2012: Our surveys had indicated big demand coming up (70% of 250 target market respondents had said they would buy the product), our alpha test group gave us thumbs up, and in-depth 1 on 1 interviews also showed that we're on to something. People really liked the idea!

So on the first day, we stacked the veggies and herbs, bought from wholesalers (not from supermarkets, pah! We were taking advantage of economies of scale!). We expected a torrent of orders.

The torrent ended up being a trickle and we got knocked out of La-La land back into reality. On the first day, we had exactly… 3 orders. 12 orders in the first week.

This will be the number one lesson I will never forget and the absolute key to understanding Dinnr's failure—we were not

solving anyone's problem. I should have found that out in my initial market research, especially in my 1–1 interviews.

However, we committed the big mistake of presenting people with the idea and asking them if they liked it and would buy it. And when people said yes, WE thought they meant "launch it and I will buy".

In reality, they meant…

"I'm not entirely excluding the possibility that one day, when Ocado trucks run out of gas, supermarket doors get blocked by red-hot lava and restaurant waiters will, due to a mysterious leak of radioactive fumes emanating from commercial kitchen equipment, all be zombified and eat patrons' brains, yes, in that case I might be tempted to purchase a trial product from you. Once. Then I'll take a risk with the zombies."

There never was a need for a service like ours in the UK in the segment that we picked (urban professionals, couples without kids or with one baby max). And I deluded myself in thinking that there was, because I was eager to start my own thing and thought that this was something that I myself would use.

So where does this discrepancy come from? Why are people enthusiastic about the idea but then end up not buying? The problem is actually quite interesting. People give positive feedback on a new business idea for mainly two reasons:

a) People want to make you, the founder, happy.

I was aware of this risk at the time, and reminded everyone in the 1–1 interviews to be 100% honest with me. And in the end, I doubt anyone tried to flatter me. BY FAR BIGGER is the following, and at the time completely hidden to me, issue:

b) People are too optimistic about their future behaviour.

Especially when it's about a product that has an aspirational halo around it. Dinnr was about learning to cook, eating healthier and home cooking — all things that most of us would agree are worthy, wholesome goals worth aspiring to. So, of course, the "new year's resolution" effect kicks in — I genuinely resolve to be a better human being (and use Dinnr).

And as we all know, new year's resolutions are made for breaking.

Side note: An interesting corollary of this software bug of the human mind is the "Such-A-Good-Idea Delusion" after the fact. You don't know how many people, when they heard that we're out of business, appeared genuinely sad and lamented "it was SUCH a good idea".

I tried not to sound passive-aggressive when I asked "well if it was such a good idea, why didn't you use it?". And then people went "well, you know, I'm too busy to cook" or "oh, yea, I guess you're right…".

By that time, I found it funny: They got the lesson for free — I had to learn the hard way over 1.5 years. I myself was deluded by the promise of "such a good idea, and so useful!" into starting it in the first place, so

I was the main victim (and perpetrator) of this mind bug.

Not every good idea is a good business. Duh. How does one detect the danger of being in "too optimistic about my own future behaviour territory"?

You're being told by the prospective customer, after you explain the product: "…yeeeeeeah [the slow, hesitant kind of yeeeeah that shows that the usefulness of the product is slowly dawning on them], that is really good…

That is really useful" with an expression on their face that indicates "I have not really thought about that but what you're saying makes total sense and it's SO TRUE!".

This is a red flag because if they haven't really thought about it, it means that it's not high priority enough for them to bother.

You learn that you're not really solving a big enough problem for this particular customer but they still compliment you about it. What I mean with "big enough problem" is simple: Is the fact that they sometimes cook bland repetitive recipes or eat takeaway food an actual pain point for the customer? One that is so big that they would actually pay to get a solution

for it? The answer in Dinnr's case is a resounding NO, of course. In hindsight.

People compliment you on the idea because they believe it will be so useful for peopleother than themselves—I.e., they get into advisor mode. "I'm not much of a cook, so I might not be the best target audience for you, but this will be SO useful for young mothers…".

Intellectually, it's clear to you in the moment that this is nothing but a hypothesis worth testing. However, emotionally, you get this warm fuzzy feeling "it'll be SO useful to young mothers!" that further adds a hue of rose-tint to the glasses that are slowly forming on your nose. An actual conversation with a young mum further down the line invariably gets tainted with this optimism that makes it easier to ignore warning signs in her feedback. Or, worse, you don't even conduct the interview with the young mum in the first place because you have testimony from someone else about them!

You must never, ever, pitch the product to the customer and ask for their feedback. Instead, the conversation serves the purpose of ascertaining whether the lack of recipe diversity and frustration with takeaway food or ready meals is a big enough problem for which I offer a solution that they would pay for. To reach this conclusion, don't ask about their future intentions ("would you buy this?"), but do ask about PAST behaviour: When was the last time they cooked something new? Do they dislike takeaway food? Do they feel bad about food wastage/spoilage? If so, what are they doing to minimize it? If nothing, it's not a problem waiting for a solution.

This is a topic superbly explained in Rob Fitzpatrick's (@robfitz) great book "The Mom Test". It's definitely the most important business book I read in the last few years. It's right up there with "The Lean Startup". I can't recommend it enough. Conduct customer research without it at your peril.

The Mom Test is the key concept that clarified my thinking over the 2013/14 New Year period and helped me come to the conclusion that I had to pull the plug asap.

Thinking back, what could I have done to correct the course I was on in those early stages? One of the things the author recommends is to ask for commitment straight away. Simply telling my enthusiastic interview partners "ok, give me £20 and I will deliver you a prototype next Tuesday" would have yielded lots of interesting insights. I now think that few would have actually done it. Why am I saying that? Out of 10 interview partners, 7 liked the idea. But when we actually launched, only two became customers, weeks and months later, respectively.

This turns out to be the original sin of Dinnr—there never was an opportunity. And whatever we did later to try to breathe life into it (iterating on the website, different marketing tactics) was akin to giving aspirin to a deathbed patient.

To wrap up this long chapter, the #1 learning for me was that my company Dinnr was a classic case of a solution looking for a problem. I conducted market research superficially, let compliments go to my head, and assumed that the problems I was solving were big enough that people would pay money for the solution.

Instead, I should have asked for commitment on the spot.

Lesson #2. The fact you haven't started a company before is not a reason to go with something easy

As background to this one, a biographic note is worth mentioning: Growing up in Austria, my brain had been marinating for years and years in a meme that was pervasive in my family, school and overall culture. That meme says

"To start a line of work, you have to dedicate a lot of time to studying it and preparing for it. And then, one day in the far future, you'll be allowed to actually DO it." With "studying" and "preparing", this meme means lots of time spent by yourself, immersed in books and anticipating the time when you may step into the limelight.

This is how the Austrian school system works. You study long and hard and THEN the big day comes. EXAM time! Performance time! You have worked hard, now SHOW what you got. Your hard preparation time will be rewarded.

Contrast this to the anglo-saxon education model that I had the privilege to experience at London Business School. There, you are constantly being evaluated (class participation is part of your grade), very often you have to present in class (in contrast, I presented ONCE in five years of law studies in Vienna), you work in groups, and you run practical projects with actual companies (for me, at least four such projects at LBS in 18 months, zero in Vienna in five years).

As a result, this education system has much higher chances of producing people who actively DO rather than just absorb.

People who are biased towards action and people who will apply their knowledge rather than regurgitate it at exam time, people who learn by doing rather than bookworms who leave uni at 30 without any practical experience. No surprise that the English-speaking world has a higher density of entrepreneurs (I realize I am a bit negative here and possibly not doing full justice to the differences between continental Europe and the UK, but I invite anyone to consider how many of your friends who wanted to go study abroad ended up choosing Austria or Germany. Case closed.)

An embodiment of the practical, meme-shattering approach was my MBA classmate and now very good friend John. He had a consulting background and wanted to get into Venture Capital. So before actually moving to London to do the MBA, he contacted several school alumni who worked in VC, lined up coffee chats with them and offered himself to work with them for free to learn.

Sounds completely normal to you, right? For me, that was nothing short of revolutionary. My Austrian meme-infused brain screamed "Are you crazy? You don't KNOW this stuff! You did consulting and not VC. You first have to take at least a course, better a concentration on VC in the MBA and THEN you can possibly apply for a job. How on earth do you think you'll be useful to them?" Well, turned out, he was pretty useful to them and subsequently got a job at one of the most prestigious VC firms in London.

Now, memes have the nasty habit of hiding in crevasses of one's mind even when you think you've purged them. In my case, it came back in the form of the following concept that I had in my head when starting Dinnr:

It's a good enough business to run as my first startup. It's relatively simple (e-commerce), I understand the value proposition, I understand the customer. It won't be a billion dollar business, but it's absolutely realistic that I could sell it for a few million in a few years. I will get very valuable experience at a low risk which I will then be able to use for future, bigger, projects. Can you see the Evil Meme hiding in there? If you rephrase the above it comes into full view:

You can't do anything too complex because you don't have the expertise. The lesson is: All that matters is that I identify an opportunity. If I need market expertise, I will find someone or will learn it myself. If I need technical help, I will find someone or learn it myself. This is what evenings and weekends are for, when in a corporate job. Expertise in a domain helps of course and gives me more credibility with investors. But examples abound where entrepreneurs succeeded outside of their original domain.

You can develop yourself towards stepping up to the opportunity. But you can't conjure up an opportunity to suit your current skill level.

Lesson #3: A concept's success elsewhere is a small nudge, not half the journey.

A good route to starting a business is to take what has worked abroad and transplant it to wherever you are. If the Samwer brothers can make a $3bn business of this blueprint, I can copy the copiers on a small scale.

Or so I thought.

I thought that because businesses like Dinnr have sprung up everywhere around Scandinavia, it will be a breeze to start something similar in London. After all, it's not like there is a vast cultural chasm between these two — it's not, say, Japan vs UK. And it wouldn't be the first time that a Swedish product conquers the world, right? Think Abba, IKEA and Spotify and the pervasive ubiquity ofrotten herring on dinner tables around the globe (ok, maybe not everything has been a hit).

What I didn't expect is that success in one market shouldn't be anything but a little nudge, and not contribute more than maybe 1% to the entrepreneur's decision-making whether to start something or not. The market research still needs to be done as thoroughly as if there hadn't been market proof elsewhere.

Apart from lacking a sophisticated online supermarket system like the UK, Scandinavia has a few cultural traits that make people much more susceptible to Dinnr-like services: As I've been told (after launch, unfortunately) by expats living in Sweden, the Swedes are generally much more attached to their routines than Brits. You can easily ask a Londoner at work if they'll have a drink with the team after work. Ask a Scandi and chances are you'll have to schedule it a few weeks in advance. They also have a much stronger "eat at home with your family" culture.

It was one of the main things that kept me going: It's working elsewhere so goddamit it must work here as well!

The lesson is: No, it doesn't have to work in country 2 only because it works in country 1.

Jessica's Recipe Bag (part of Linas Matkasse, the Swedish juggernaut in the space) has withdrawn from the UK market in early 2014, Housebites have packed it in in June 2014 and The Recipe Kit are also apparently not trading any more. Yes, Gousto and Hellofresh are around but they are going after a different market segment than us. Generally, my impression is that this is not a product for the UK.

Lesson #4: Be your own worst critic, especially early on

Although I would say that I am reasonably well connected, have many startup CEO friends, go to events and am regularly speaking to mentors, no one ever sat down with me and systematically challenged me about the Dinnr concept.

I did have a few people who were very negative about it but no one of them actually gave me good reasons for why it wouldn't work. The main themes were based on the premise "I wouldn't buy this myself, so it won't work." which doesn't help much—is this just a naysayer or does this feedback tell me something about a fundamentally flawed premise that I'm not seeing?

I really would have needed a critic who said:

"Look, you have the following problems":

1. Show me your market research. What questions did you ask? How did you conclude it's a good opportunity? Wait—are you telling me that in ALL your conversations you mentioned the product and asked for feedback? Well that's a HUGE problem, you know that, right?

2. You're saying it works very well in Sweden. You do realize there's no comprehensive online supermarket in Sweden, right? Is THIS maybe the reason this concept works in Sweden?

3. What are your assumptions about the customer's problem? That food waste is an issue for them? Wait—you're telling me that throwing away the occasional old bunch of spring onions or moldy yoghurt is an actual pain point? Isn't it rather something like a… minor annoyance not even worth thinking about?" etc. etc.

I was hoping it would happen during the pitches I gave to investors. But the rejection was much more vague "hmmm—not sure there's a market here." "Not convinced about how you'll scale it" etc., or, my favourite, "unfortunately we will not be able to move forward" (never actually said but only written in generic rejection emails.)

Having such detailed feedback wouldn't require diving into the data, but having someone spend an hour stress-testing my thinking and assumptions would have been gold dust and could have prevented a lot of effort wasted. I'm sure I'd have gotten that had Dinnr been accepted into accelerator programmes such as Seedcamp, TechStars or Wayra. But absent a full-time co-founder, we weren't eligible for the former two and narrowly missed the entry into the latter.

I also realize that I did not take advantage of the fact that, through crowdfunding with Seedrs, I had access to many very smart people who invested in Dinnr. I'm sure that had I asked 10 of them to spend an hour stress-testing my assumptions, 2–3 would have agreed and I could have gotten something out of it.

No Deus ex Machina will come and challenge you in the right spots. This is why you got your own experience, both theoretical (those startup books DO help!) and practical for. You have to do it yourself.

The lesson for me is: Especially in the early days, you need to be your own worst critic and shoot down your own idea from as many angles as possible and question everything.

Lesson #5: Run a tight ship with development and design

This is a quick one—We had offshore developers and designers who, despite coming with recommendations, didn't execute well.

As one of my first investors later told me, you must have developers on your core team in the same room. You need to be able to iterate and execute fast. There is no slower way than doing this with developers overseas, even if your company

isn't tech-heavy (such as Dinnr).

Lesson #6: Professional design will not improve the business fundamentals

In April 2013, I had a great conversation with an entrepreneur who had successfully exited a homewares business. She told me that the main problem of Dinnr at the time was our amateurish website.

Her points were:

With food, it's a lot about trust—the more professionally designed website, the more likely people will trust you and you'll keep them as customers. You have to address the senses—you want to get the impulse buyers. Improve your photography, make the pictures huge. Basically, it has to be food porn to convince customers on an ongoing basis.

At the time, I spoke to many people who gave me their opinions on Dinnr—her advice stood out to me because it brought together two points that previously had been disjointed: I didn't like the look of the website and I wasn't too happy with our repurchase rate. So naturally, finding a recipe that would solve both at once was very appealing.

However, 5 months later, after the new website was up and running and a brief flurry of new orders had come in, the repurchase rate went back to where it used to be. The hypothesis of "make it visually more appealing and people will come back more often because you address their emotions" hadn't worked out.

The conclusion is—you can't design your way out of a fundamental business flaw.

What I had overlooked at the time as well was the fact that one of her talents was design and that she was seeing the business' problems mainly in that light. When your main tool is a hammer, more problems looks like a nail.

Lesson #7: Expect results faster and attach consequences to goals not reached

Yes, every startup starts small. And I did set monthly goals. But often, we didn't reach them by quite a tall margin. The problem is that I didn't attach any consequences to not reaching goals.

The last thing I wanted to be is someone who quits early, but I should have spent more time beforehand to investigate metrics that we need to hit in order to continue. The exercise should have been thus:

- Let's get our first 100 customers.

- Then, ignore further customer acquisition and get these 100 customers to a repurchase rate of x.

- If x isn't reached by date y, abandon project.

- If x is almost reached, tweak model for a while but don't wait for too long.

Of course, this doesn't mean that after a bit of trying, you quit. But having such a set of if-thens would have helped in establishing boundaries to our plodding along. Once a goal hasn't been reached, it would have helped us to sit down and to decide that we had to change course no matter what. In retrospect, it looks a bit crazy that we didn't correct our course despite often not hitting the numbers.

The Good Bits:

Not to be only negative here: There were moments at Dinnr that were very rewarding and fun. Two stand out:

Valentine's Day 2013, we had a total of 23 orders with multiple courses. We were well prepared for it and the headquarter was buzzing with activity. On that day, we felt a real spirit of camaraderie and "we're on to something here". One of the moments you start a company for.

On Thanksgiving 2012, we received an order from a lady whose family was visiting from the US. She said that their 10 year old son had fallen ill and was in the hospital. They were ordering our product so that they could prepare the food at home and bring it to the hospital to have a Thanksgiving dinner there, with the boy. Afterwards, the lady sent us a picture that gave us goosebumps: The family are sitting around a table with the boy who apparently had contracted a serious disease. He was sitting at the head of the table, with no hair and a drip in his arm. The customer told us that we made a big difference to their day and that the boy was really happy.

For these and other such moments, for the opportunity of having worked with a fabulous team and dedicated investors, it was worth doing it all.

Even those miserable moments in January 2014, cycling through the icy rain to buy a single lemon, returning back to the dank cold warehouse, staring at the Dinnr backend, waiting in vain for orders to arrive.

Who is there to thank?

In no particular order:

- My cofounder Adil. Without his experience and encouragement it probably would have taken me another few years before I'd have taken the plunge into entrepreneurship land. He was instrumental in getting the first version of the website off the ground in record time, taught me a great deal about managing developers and designers and was my most important advisor for the 1.5 years

- A great team of people who, given their smarts, talent and great attitudes have given up other opportunities of employment and instead poured their effort into the venture, for very little or no pay. Megan, Mike, Daisy, Charlie, Toby, Enikoe, Patrick, Matt, Ludovic, Alessandra, Himani, - I hope I didn't forget anyone - thank you so much for all your hard work, ideas and energy!

- My mum, my brother and 97 people on Seedrs who invested in us, believed in the idea and the team, and often provided us with words of encouragement, advice and good probing questions.

- The Seedrs team: Jeff, Carlos, Tom, Alysia, Frank, Simon, Ricardo (and now that the team has grown, surely many more)—A stellar lineup of fantastic individuals who did so much more for me than help with the money they provided

ideas, advice, peptalk, custom, and many opportunities to meet interesting people (and some pretty cool headshots of your humble servant, courtesy of camera wiz Tom Davies!)

- Jessica Andersson, of Jessica's Recipe Bag. She greatly helped us after we had to move the business to new premises, allowing us to share their warehouse for a few weeks.

- All our customers who liked us, bought from us, gave us feedback and (most importantly!) kept coming back. You gave us hope and kept us going.

- And so many others who provided words of advice and shoulder pats! You know who you are.

I can be followed on Twitter at @mbohanes

Original article by Michael Bohanes here: https://medium.com/@michalbohanes/seven-lessons-i-learned-from-the-failure-of-my-first-startup-dinnr-c166d1cfb8b8

Startup Failure: Seismic Video

How I Failed Launching Seesmic Video - It wasn't about technology

I see many startup creators focus on the technology and the product features when they work on their product launch. I did just that when I started working on the first iteration of Seesmic in 2007.

Conversations in video show you as you are

I like online conversations and think there is no better way online to get as close as possible to reality than recording a video of yourself and sharing it.

Videos bring trust

In order to create trust online and a real community, you need to be yourself as you really are… If you follow me online and meet me in person there is a good chance that you find the same person. Many people create a fake image of themselves online, try to look smarter or better (like, I could pretend that I have a good english accent) and the result is that trust is broken as soon as whomever stumbles upon them online figures it out.

My idea was just to bring the conversations we have in person online but not in realtime so anyone can jump on the thread at any moment.

Why it failed

Sure, it was seven years ago, pre-iPhone and pre-Android, so it was ahead of its time, we had to use Adobe Flash on a browser which sucked in so many ways I can't even start to explain how bad it was. Technology would be so much better and more important all mobile today.

The main reason Seesmic Video failed is because of human factors, not because of technology

Most people are not comfortable sharing videos online. They think they look like sheeeeeet (and often do). They are shy. Recording yourself talking to a tiny camera isn't intuitive and doesn't feel right. It's for weirdos (like myself).

That's why it did not take off, it's too much of a human habit change.

I am not convinced it would take off today, 7 years after I tried, because those human issues are still there.

Now of course you could say that selfies did not really exist a few years ago and are now super popular. Video selfies might feel natural soon.

Or you could say that the solution is Snapchat, sending it private to your friends without keeping a recorded version. Sure, that solves the fact that you look like crap and are afraid of saving that version of you for the public to see forever on the Internet. Not sure.

I just played with YouTube and Vine, also with a new app called Vlogma to see how technology evolved. Not that much it seems. YouTube is too slow to post and Vine isn't made for conversations.

Do you think video conversation and video selfies will grow to become a mass phenomenon?

update: saw some good points about my post. The community that Seesmic Video created at the time was amazing, but small. It can be seen as a success that way and could have remained an interesting small service, sadly that wasn't the objective.

Original article by Loic Le Meur here: https://medium.com/@loic/how-i-failed-launching-seesmic-video-81e898dcf751

Startup Failure: Treehouse Logic

Thoughts on shutting down Treehouse Logic

We recently shut down Treehouse Logic and I thought I'd share some insight on where we went wrong. Treehouse was a visual configurator platform company that helped retailers and brands quickly build product customization experiences on their ecommerce sites. We originally set out to be "the SurveyMonkey of configurators."

First I will mention the irony that in most ways we accomplished exactly what we set out to do. In 2009 we sat down with Timbuk2 and learned about the weaknesses of first-generation Flash configurators. Our theory was that the SMB brands of the world were ready to include configurators in their marketing strategy, but at lower prices. We identified the main problems that retailers had with legacy configurators. They were

- Performance (Flash can be slow. Server-based configurators require browser refresh)
- Cost – custom code
- Not editable
- Not mobile friendly
- Not social friendly, ie not sharable
- Not guided – no rules (Most customers abandon configurators because they get creatively frustrated)
- Time to market – take 9 months to ship
- Difficult to integrate with ecommerce systems

Here's how we solved those problems

- Performance (Client-based configurator, Javascript)
- Cost – a fraction the cost of custom coding our boutique agencies.
- Editable - Our clients could edit their configurator daily.
- Mobile friendly - HTML, not Flash
- Social friendly - every configuration is defined by the URL parameters
- Guided - a rules engine for compatibility and constraints. Customer proof.
- Time to market – In some cases 3 weeks to ship
- Integration - easily done by passing URL parameters. No need for APIs.

These legacy problems lead to really low sales conversion rates for custom-coded configurators. The data showed that we had accomplished what we set out to do: Some of our Treehouse Logic clients saw up to 6% conversion rates - as good or better than Amazon! Sadly, solving those customer pain points didn't lead to a sustainable, scalable business as I'll explain below.

Business model

Most of our potential clients wanted a self-service dashboard, like Wordpress for blogging or Surveymonkey for surveys. They were running a Shopify or Magento shop (or BigCommerce or 3Dcart) and wanted to handle almost everything themselves. They expect a configurator plug-in to be free or $200 and they expect it to let them easily build and brand any kind of configurator. They didn't want expensive agency services. Rather, they wanted a self-serve platform like WordPress.

Everyone expects free or cheap plug-ins on top of a framework, but we had accidentally built a services company that delivered configurators based on our internal customization platform. We couldn't find a way to standardize customization so that a template or self-service CMS with limited options made sense for a broad enough audience. The configurator landscape is too diverse and fragmented, unlike the survey market (SurveyMonkey) or the blog market (Wordpress). We chose flexibility and feature-rich over constrained and marketing-friendly. One reason flexibility seemed like a good choice in the beginning was that above all, brands are concerned with the integrity of their brand. They won't accept a templatized solution for something so strategic to their brand identify.

A good analogy is that we built extremely flexible kitchens that can be assembled very quickly for any type of restaurant. But clients also need chefs, waiters, restaurant managers, dishwashers and someone to buy their groceries. Clients just want a finished restaurant and they rarely have all the skills to put it all together. But of course a restaurant-in-box model

won't work for most restaurants either, because each restaurant has their own kitchen experience in mind.

Channel

Related, we wanted to hitch our horse to an ecommerce platform like Magento or BigCommerce. Since we didn't have a full agency team or sales force, finding the right channel was important. Ecommerce platforms are still very fragmented. There are 200 shopping cart solutions out there and many shops still develop their own from scratch.

We really thought partnering with the right Magneto dev shop was the right model for us. We just couldn't make the economics work. Those dev shops don't want to pay for a platform and split the client budget. In the end they'd rather 'fake it' by stitching together a hand-coded configurator and make more money on services. Our platform only made sense to agencies if its cheap and saves them time / money.

We wanted to stay small and partner with vendors but the value chain was too complex and we didn't have any desire to build ourselves up as a full-service agency since we are product developers at heart, not creatives services guys.

Customer acquisition

We crushed the free SEO piece (our blog was our secret sauce, in terms of marketing). We literally got 1 lead per day, even after we shut down. Inbound marketing is great for start-ups with no budget. It's the opposite of the enterprise sales model where you have high paid, dedicated sales people pursuing big companies. But, the quality of leads was really bad. We get the mom-and-pop shops that see what NIKE did with design-a-shoe (for $1M) and want to imitate it for $200. "My wife and I want to sell custom jewelry online…"

If we are going to serve small businesses, we need a way to let these SMBs on-board quickly with minimal interaction. Many of these leads were a good fit but we still struggled to ship the configurator quickly. More on that later.

Related, we confused interest with traction. We got a lot of attention from industry thought leaders and customers, but interest is not the same as traction. I heard "you are exactly what I'm looking for!" almost daily which was encouraging, but often didn't lead to money in the bank in the short-term. Traction is when you are successfully converting interest to revenue, and learning how to scale your business (ie reduce costs and increase revenue)

Big companies are a pain also

We had a few big name retailers lined up, at least initiatlly. But that doesn't mean that they had high budget or prioritized the configurator. It usually means long sales cycle and lots of red tape which we couldn't afford to deal with. They will bleed you dry. It's tempting to think that all we need is a few big clients with big budgets and we will be well paid and not have to deal with a high volume of difficult small clients, but it's a myth. The best model is to on-board a high volume of clients with as little manual interaction as possible. More on that later.

Build vs. Buy

Many potential customers are convinced that their technology is their key strategic asset, so they build everything from scratch. Even small fashion startups with 3 people think of their technology as their primary asset. Any company that is serious about customization wants to own the experience and technology. They think technology is their IP. Our pitch was that it's better to let a technology company handle the technology. They are in the fashion business, not the technology business.

We had tons of potential clients who said "I wish I had met you 1 year ago!" Most clients who are dealing with rubbish code end up deciding to keep babysitting it rather than ripping and replacing it. Some of these clients even got funded with a rubbish configurator! The reason is that the configurator turns out to be mostly conceptual, ie "look, customers can design it themselves!" rather than actually being the backbone of the business.

Pricing

Someone in India can always do it for $5000

First-gen configurators like Nike ID and Timbuk2 used to cost $500K to build, with Treehouse Logic platform it can be

done for $40-50K. That's all-in, including design, CSS, images, and ecommerce integration. We couldn't land deals at $50K so we reduced our price to $5K plus $200/mo hosting / support fees. Most customers see the monthly hosting payment as an invisible tax that they want to avoid, even if we position it as "licensing and support." Our system was a hosted SaaS solution (likeCRM and Cloud Computing To Grow Your Business - Salesforce.com) but clients still think of it as a website that can be delivered for a one-time fee.

Our main pitch was that using a platform is smarter than custom coding. Custom coding is a pain in the ass and prone to complication. Still, many clients will outsource to India because of the low sticker price and learn the hard way. Most clients still choose cheap over good.

Most configurators are shipped and left alone in the corner of a website, ironically. It's always a heart-breaker for the client when they realize how hard it is to add products or edit product data in their configurator, but they never realize that until it's too late. Or they just don't intend to edit it. Or they never spend any money promoting it and it goes unused and is left to die.

Mission critical apps

Although everyone and their brother wants a design-your-own section of their ecommerce website, it's rarely strategic compared to other hot ecommerce trends like social commerce (Pinterest / Instagram) or the fundamentals of ecommerce 1.0 (price, convenience, selection), or being on Fab.com | daily design for everyone, or having a flash sales deal, or good email marketing, or having a great Facebook engagement strategy. Often, configurators are just 'tick boxes' that set their brand apart from Amazon, give customers something to engage with, but they are not optimized for sales, nor they aren't fundamental to the definition of the brand.

Consider ecommerce for a moment. Usually there are two ways to find products on an Amazon-like catalog site: Search and browse. Customization adds a third way to find and buy a product. Pretty soon you start asking yourself why there are so many ways to buy the product, shouldn't there be just one simple, guided, branded way to discover, research, select, and checkout (ie shop?)

I've always argued that configurators are really engagement tools, so they will evolve into social commerce tools where shareability and aspirational shopping are the main ingredients. Ford is a great example, their mustang configurator is a Facebook app that means to power a contest. It's not meant to generate sales. Burberry said the exact same thing about their custom trench coat tool. It's just a way to let you experience Burberry.

Originally, configurators were sales tool for complex B2B products (like tractors and firetrucks), but enterprise configuration is a dying industry, and there are dozens of old fashioned enterprise companies from the 90s that do ugly drop-down window "sales quote tools."

No competitors

It's probably a bad sign when you don't have any direct competitors. ;) Our main competitors were · Custom coded configurators. · Digital agencies · Sales quote configurator software

I wouldn't consider Adobe Scene7 a competitor since they mostly focus on image creation. Their system was never intended for ecommerce.

Branding

One reason generizing the configurator is problematic is that each brand wants a branded experience. The customization experience IS the brand. The branding of the customizer is just as important as the functionality or its contents. This is also why marketplaces like Zazzle, CafePress, and eBay struggle; they want brands to come to their domain, whereas on on-site, branded configurator is what most brands really want. Going off-domain makes sense for selling pre-defined products like books or DVD players.

Should we focus on the configurator engine or offer full-service?

We don't do windows

Probably the biggest reason we failed was "the project management nightmare." Even if a client verbally confirms they

want to work with us, we need to scope the entire project to make sure there are no custom features (there is always at least one feature we need to add to our platform for them), then we need to help them figure out

-their images -their CSS -their ecommerce solution

…all of which we don't handle ourselves because we are agnostic about it. You can imagine how this pitch goes over: "Yes, we're ready to start, just get your thousands of images up on FTP and put the links in the XML, hire a CSS specialist, and an ecommerce developer and we're ready do go!"

The reality is that the client expects full service and doesn't have a low level dev / creative team to complement our "unskinned configurator." It's very confusing to them when we recommend 2-3 other vendors and form a virtual agency. And the cost creeps up.

And not to point the finger but most of the time when we did bring in outside vendors they dropped the ball because the project wasn't dead simple. I don't think we ever used the same CSS, design, ecommerce, or image creation vendor twice because they all suck.

Configurators are too complex to standardize

Each configurator is different and configurators can be very complex. We had an internal joke: "baby cribs are rocket science" because we had a client that sells baby cribs and they were really complex.

Keep in mind that I'm not talking about technically complex. I'm talking about lots of data and moving parts complex. We rarely hit any technical speed bumps. Our technology was amazing, but its a good lesson to learn: technology doesn't necessarily solve the right problem for the customer.

We were unable to platformize the configurator space, meaning we were unable to build one standard configurator that applies to many clients. This is mostly because each project is different. We built in an amazing amount of flexibility, but there is not "just upload" your content and you're all ready to go" model.

Also I would mention that some of our client projects were not finished after as long as TWO YEARS. It's not that the project was complex or that we were the bottleneck, it's that we were expecting something from the client like images or a review on their end or for them to have someone CSS style it. Often the project was out of our control after we did the heavy-lifting, and clients just left it hanging. It's so frustrating to not control the button that pushes the build live.

Market fit

Many potential customers wanted personalization, not customization. As many as 50% of our leads are inquiries for personalization, ie text overlay or image upload, which we don't do. This is the online design tools vs. product configuration issue. Our platform was a rules engine at its core.

We really build a platform for "configurator engineers," which was another fatal flaw. One of our founders was a configurator support engineer at a large IT company and we mistakenly thought if we served his need we would serve the market's need. What we should have done is served the self-service small business, similar to the Magento do-it-yourself model. Every time I brought this up we were reminded how complex configurators are and how we weren't close to have a real self-serve platform.

We spent so much time building a platform that we were convinced we could charge clients "for access to our platform." The reality is that they don't think of it as a platform, they think of it as a website. The platform is just a means to an end.

The wrong problem

We mistakenly thought the primary problem to solve was guidance. Having a design-it-any-way configurator gives the user too much choice and leads to frustration, so we thought having rules would solve that problem. It turns out rules can be too constraining. Most clients just want a color picker configurator with no filters or rules.

Guidance is still very much strategically important in the shopping market and it has manifested itself most recently in ecommerce models as curation. See Fab.com | daily design for everyone as an example of "hand-picked by design

experts." Keep in mind that customers want great products, not necessarily custom products. And convenience. Customizing a complex product (like a dress shirt) is a lot of work. Never make your customers do hard work.

The key important lesson in the product customization market is that consumers THINK they want customization but they quickly get frustrated and seek out expert recommendations instead of actually building from scratch. We learned this in our first meeting with Timbuk2. Most customers flock to the configurator, get excited about the power of the Internet, then abandon the design process and buy a standard product that was designed by an expert. Your expertise is more valuable than anything. It's your story, it's your brand.

Now this doesn't mean that customization isn't adding value to the brand experience. It is. I still believe that customization will part of EVERY shopping flow, but it won't be build-it-from-scratch customization, it will be light modifications right before check out. Empower the user but make the overall experience simple and fun.

A few non-issues that advisors thought would be critical

Integration

I should mention that the most challenging technical with configurators in the 90s was system integration. The main headache was the manufacturing process, ie how to get the ordering system to talk to the backend ERP system. We never had any issue with integration. It was a non-issue for us. As long as the ecommerce system could receive an order via HTTP from an external source we could integrate. The configurator talks to the ecommerce system which talks to the ERP system. I rarely saw any issue with this unless clients were using something bloated and archaic like NetSuite.

Enterprise grade

Many clients were concerned that we weren't "enterprise grade" but I've really found that whole discussion to be a misconception. Our system was hosted in the Google App Engine so we are just as enterprise-grade as Google or Amazon. In the end we passed any Enterprise grade questions with flying colors.

Related, many clients would see how javascript based configurator and assume it was not enterprise grade or secure because it's not server-based. In reality all client data is obfuscated and secure. Sometimes the visualization and impressive design gave engineers the impression that the tool was not enterprise software. Not sure why.

Sales nightmares

I'll add a few horror stories that we experienced on the sales side

Long sales cycle

Sometimes we had to do TWENTY mockup versions before we could get a contract signed. We had to define the entire project before either side could commit to it. Clients are obsessed with the user experience, which makes sense, and nailing the configurator flow is top priority.

Exclusivity

One client loved everything we offered and was ready to sign but wanted exclusivity for his industry in a regional area. In the end we just couldn't accept those terms since we were a platform that depends on enabling any viable client. Startups can't afford exclusivity.

Downtime insurance

One client wanted us to sign terms that said that if the configurator went down we would pay them the equivalent of lost sales for the time the configurator was down. Again, not a good fit for a startup.

Beta customers

We gave away our services for free to several early beta customers. Some of them never shipped because they couldn't put together all the missing pieces like images and CSS styling.

Bottom line

Startups fail when they are not solving a market problem. We were not solving a large enough problem that we could universally serve with a scalable solution. We had great technology, great data on shopping behavior, great reputation as a though leader, great expertise, great advisors, etc, but what we didn't have was technology or business model that solved a pain point in a scalable way.

Original articles here: http://www.quora.com/Why-do-customization-startups-fail

Startup Failure: Backchat

The End of a Great Experience

I want to start this final note by saying that is it with the upmost gratitude and respect that I have had the privilege of getting to the point where I need to write this post.

A short two years ago I would have never dreamed of being able to have the chance to build a service that could help enrich peoples lives is still surprising in my mind. When I started with my father, Uri, two years on YouTell it wasn't really apparent the new field of social interaction we had discovered. At the time anonymity had been on a downswing, through the years we launched YouTell and Backchat two anonymous services in social.

YouTell led the way with anonymous feedback from the people you care about to get real honesty without letting identity get in the way of the truth. With the discovery of how direct feedback could also be applied to messaging which led to the first version of Backchat on iOS, then soon after on Android.

The amount of support that our users gave us and the memories and moments created still ring in my head until today. We were able to be part of the beginning of a new chapter in social networking which is picking up the pace everyday with companies Secret and Whisper.

Unfortunately we were not able to adapt fast enough to changing market and product conditions which quickly began to show in usage metrics. With a feed product on the horizon all looked well until our funding began to dry up. Deals fell through leaving me in the difficult position to close Backchat and YouTell.

To our devoted users I know this is disappointing and I hope you can find similar experiences on other platforms in the future. To those who I met through email for brief moments, or those whose advice helped guide me through difficult decisions and situations a huge thank you to you. Most importantly I want to thank the team that helped make this all possible. Our conversations about hockey while waiting for each new build to ship or deliberating over button placement and font sizes. These moments will forever share a special place in my heart.

All of Backchat and YouTell have been shut down, they will remain in the app store but server-side functionality has been disabled.

While this has been tough news to come to terms with I understand it happens and I'm glad I was able to be there to be part of it. I will try to answer some questions if there are any but I thought it would be appropriate to give some closure to the company and its products.

The most special of all the thank you's I can give is to my dad, who was the single biggest influence to me and the co-founder of both companies and deserves equal recognition for his efforts.

I hope to be at the startup game once again sometime but I don't know when or where or for what, but until then this is my last post.

-Daniel Singer

Original article here: http://blog.getbackchat.com/post/88313702290/the-end-of-a-great-experience

Patient Communicator

Why Patient Communicator Failed

This is a post I meant to write a lot sooner when it was more relevant. But first, a little background:

My father is a primary care doctor. In 2009 he asked me to help build a product that would enable his patients to access him online rather than through the phone. Patient Communicator was basically a patient portal + CRM for doctors + practice management system and it completely transformed my father's practice. (Read his profile in Forbes.) Then in January 2012, I was part of the first class at Blueprint Health, a healthcare tech incubator that was part of the Techstars network, and my co-founder Larry Cobrin and I spend the next 3 months trying to commercialize the product.

(Sidenote: Blueprint Health's third class, Winter 2013, will start on Monday, Jan 14th, down in the SoHo office. Brad and Mat have done a really good job of finding exceptional companies in the healthcare space.)

The sum total of our efforts - which included hundreds of cold-calls, tens of thousands of emails (I was invited to a "top 20 customers" dinner with TK from Tout), seminars, an appearance on FOX News Live, ads, reaching out to our personal networks as well as the Blueprint Health mentor network - was one paying doctor (who later went out of business) and a handshake deal to partner with a small 20-year-old EMR.

We spent another 3 months trying to raise money, meeting with upwards of 40 investors, but decided this wasn't going to work for us. Here are the reasons we decided to move on:

Business > Product: PC began as a product for my father's medical practice. Plain and simple, I never assessed the market need for a patient portal. It's extraordinarily difficult to take a product that was built perfectly for a particular user and commercialize that into a broader market. I remember constantly asking my father over the 2 years of iterative development "Dad, is feature xyz something that other PCPs would want in their office? Is this something other doctors have to deal with?" The answer was usually yes and this fueled my thinking that what we were building would have wide appeal. Meanwhile, I never really spent time pitching to other doctors to see if anyone would pull the trigger and use PC. The other companies at BPH had all started with assessing a larger need or pain point within an industry, which explained why many of them hadn't yet built a product.

The impenetrable black box: I learned that typical doctors offices operate within a black box. The doctor *will not* change their infrastructure, no matter what. They drag their feet on any new technology or changes that impact their workflow. They insist on keeping Betty, the 80-year-old secretary, their rotary phones, their paper charts, their fax machines, and their 10-person staff. A PC sale meant we would puncture that box and shake things up internally and externally: the doctor would need to adjust behavior, the staff would use a new system, the patients would have to learn a new website.

Turns out, that doesn't happen without a fight… or, in the case of EMRs, without massive government subsidies to encourage doctors to join the 21st century with electronic medical records. We weren't going to win many of them over. As a sales guy told me, this is a ground war. The secret to winning? Sell a service that feeds their black box and has a clear payoff. ZocDoc, when it started, didn't touch the office. They just said "pay us $250/mo and we'll hand over 30 or 40 new patients."

If the doctor wanted to process those patients with paper forms and fax machines with high overhead, that was their problem. Same deal with Five O clock records, which doesn't cost the office anything and instead sits just outside the black box operation, between them and law firms and insurance companies who want to request medical records. They manage the request and charges and take a piece of the action. Betty will still have to pull Jon Smith's medical chart and sit at a fax machine for 12 hours (this is an "inside the black box" task) but at least the request and charges will be handled by this website.

My father's operation is exceptional, not typical: Unlike virtually every doctor I've spoken with, my father welcomes new technology and is endlessly trying to optimize his operation. He looks to optimize everything and will happily try and switch to a new billing system if it provides more value. A typical office will use the same technology for years, and is scared or unwilling to change. My father thinks rationally and pragmatically about how to reduce costs.

He'll learn new technology, reduce staff, and shifted to doing more "secretarial" work. I had based a lot of assumptions

about the market on my father's atypical operation and atypical outlook on technology as a whole. Doctors sit squarely at the back of the technology adoption curve: What we found in the market is that doctors are generally "laggards" in technology adoption. They might buy the latest iPad but that doesn't translate to operational efficiency or the desire or willingness to adapt their workflow.

(I know, this is starting to sound redundant.)

Patient Communicator relied on doctors who understood that increased engagement with patients meant more revenue, better outcomes and enormous savings.

The trouble is, doctors almost never change their operational processes. They clutch their fax machines and rely on staff to field phone calls, and (at least in my father's experience) would prefer to shut down their practices or join a less fulfilling group practice than take control and update their operation to best practices.

Doctors are nearly impossible to access: Unless you've got a clear distribution channel or partnerships, good luck getting the attention of doctors. Pharma, insurance, hospitals, group practices, etc are all spending big bucks trying to get the attention of doctors.

Doctors are routinely being "sold to" by a ton of companies and services so you need a way to cut through the noise and get the ear of doctors. But doctors are used to getting Hawaii golf trips, catered lunches and the like (maybe less than before, but the mindset is still there). We spend day and night trying to have conversations with doctors and it was nearly impossible. Persistence is, of course, important, but at what cost? Even if we raised money, how would that change this obstacle? (EMRs partnerships…?)

Patient portals are commoditized already: (Answer to previous question: Nope!) Most of the EMRs are already working towards similar systems. PC is probably 5 years ahead of any of these patient portals (we looked at dozens of them, they are all pitiful), but that doesn't really matter. EMRs view patient portals as a trivial thing to add (usually with poor UX), and doctors generally think they should be free or incredibly cheap.

But show me a doctor who uses a patient portal (and isn't part of a larger group that requires it, and isn't a dentist) and I'll show you 5,000 doctors who doesn't. Yet we found that doctors and the market in general assigned an extremely low value to such a product.

So: they don't use it, but they value it at about $50-100 per month. We talked to dozens of EMRs and they had similar outlooks. One EMR we almost partnered with insisted on a fixed monthly fee of $20!

Meanwhile, PC was the reason my father is still in business, operating with no employees at a 15% overhead and providing a superior service to patients at no additional cost. I was certain PC would be valued at $200-400/mo, but I wasn't even close.

Crowded market: We started as a product and built Patient Communicator in a vacuum. I should have told my father to try a few products and see if something on the market (in 2008) would have done at least 75% of what he wanted. If we did that, we probably would never have built PC. And if we still decided to build it, we would have been extremely well informed on what our differentiating factors were. After all, my father would have tried a few products with his patients, seen what worked and what didn't work and what they were lacking, and therein we'd have discovered a functionality gap that might have been worth pursuing. If we had proceeded with development in this way, we wouldn't have struggled so much to explain to investors and EMRs why PC is different from all the other patient portals on the market. With such an approach, we would have understood exactly how we fit into the ecosystem. Or, more importantly, we might not have even attempted to build it in the first place!

Lack of passion: I am not proud to say this but after dealing with all the above issues, I became disillusioned with the state of health care and innovation. At first, I was fired up because I saw first hand the unbelievable impact the product had on my father's practice and his 1,200 patients.

That model - a small practice, high touch, low overhead, and the use of technology to improve operational efficiency and delivery of care - was what drove me into Blueprint Health and to make the decision to go full time on Patient Communicator. But I realized that many of the true money-making businesses in healthcare really aren't about optimizing delivery of primary care.

This is a longer discussion but I realized, essentially, that we had no customers because no one was really interested in the model we were pitching. Doctors want more patients, not an efficient office.

Maybe we could have made it work, even with the limited funding and difficult customer acquisition process. But why? That probably sounds like giving up, but if you consider that you really only have a limited amount of time and money to dedicate to your startup, why not pick a startup/idea/project that is most likely to succeed?

No CTO: (This is last on the list because I think it's the least important reason on why we failed.) We needed a full-time, on site CTO to iterate and improve the product.

When we found doctors who were interested, we had to weigh all improvements and tweaks against the consulting cost. That is not how a successful tech startup should run. You need an owner to power through code daily to constantly improve the product. You need to experiment with features, not only choose a priori the ones that have the best return on investment. Like most startups, we were not smart enough to make those calls.

Startups will have a challenge funding a lot of their growth initiatives in the early days, but not funding product development is always going to be a drag on success.

I hope this is helpful to healthcare entrepreneurs. I am generally discouraged by the industry but I know a lot of talented people are trying hard to improve things and that's good for everyone.

Postscript:

The effects of the stimulus money to get physicians to adopt EMRs is worth noting. Doctors and hospitals got on board with EMRs to get the money but electronic systems have not had any major impact on controlling costs or improving efficiency. Article after article says the same thing: EMRs have been optimized for billing aggressively and justifying these bills with "templated" encyclopedic notes that often are fabricated through excessive copy-pasting.

One consequence of this is that Medicare Claims have gone up. Now Medicare will respond with even more aggressive punitive measures and rules. A lot of this will be a race to the bottom for most while doctors who pay attention to optimizing their operation, as opposed to billing, will benefit tremendously.

One of the most frustrating things to see in healthcare is that the rapid movement of information and the potential for true collaboration to save lives just isn't happening. Maybe the healthcare entrepreneurs can move the industry in the right direction. In the meantime, I've started going to One Medical.

Original article here: http://planetjeffro.com/post/40340494649/why-patient-communicator-failed

Startup Failure: Twitpic

Twitpic is shutting down

Twitpic will be shutting down September 25th. You will be able to export all your photos and videos. We'll let everyone know when this feature is live in the next few days. (The export tool can be found at http://twitpic.com/account/settings)

This is an unexpected and hard announcement for us to make and we want to lay out what led us to this decision.

A few weeks ago Twitter contacted our legal demanding that we abandon our trademark application or risk losing access to their API. This came as a shock to us since Twitpic has been around since early 2008, and our trademark application has been in the USPTO since 2009.

Here is some backstory on the history of our trademark:

We originally filed for our trademark in 2009 and our first use in commerce dates back to February 2008 when we launched. We encountered several hurdles and difficulties in getting our trademark approved even though our first use in commerce predated other applications, but we worked through each challenge and in fact had just recently finished the last one. During the "published for opposition" phase of the trademark is when Twitter reached out to our counsel and implied we could be denied access to their API if we did not give up our mark.

Unfortunately we do not have the resources to fend off a large company like Twitter to maintain our mark which we believe whole heartedly is rightfully ours. Therefore, we have decided to shut down Twitpic.

On a personal note I (@noaheverett) want to thank you for letting us be a part of your life and helping you share your experiences over the past 6 years, it's truly been an honor. I have learned so much through running Twitpic over the years. Through the many mistakes I've made and lessons learned, to the bad days and the great days. Thank you again everyone…I will miss and cherish the days of Twitpic we had together.

Original article here: http://blog.twitpic.com/2014/09/twitpic-is-shutting-down/

Berg

Folks, I've got some news about Berg to share. We're wrapping up for this incarnation. Our partnerships and our services, they're done. A few things left, then hibernation.

Little Printer we'll run till March next year. A skeleton crew will keep it alive. We'll sell it. Or open the code. See here for more. We'd love to see LP survive.

We've not reached a sustainable business in connected products. But: There's our troop! Cultural inventions! I'm proud of this British Experimental Rocket Group.

Thank you fellow travellers, in your thousands. Behind the mountains, there are more mountains.

-Matt

Original article here: http://blog.bergcloud.com/2014/09/09/week-483/

Startup Failure: Wishberg

The Final Note - Thanks from Team Wishberg

We've spent last few years building Wishberg as a default destination for users to share their wishes & discover more interesting wishes from people around the world - things to do, things to buy, travel, adventure, experiences and so much more. The journey has been beautiful, exhilarating, exciting with lots of learning for all of us.

In July 2011, we started on this idea with Tyche'd, we failed. Our failure with Tyche'd was disheartening but we restarted our journey with Wishberg. We were humbled with the support we received from users on Wishberg - from being named among the top startups to be featured among the best products. Our simplest joys and happiest moments were to see people sharing hundreds and thousands of wishes on Wishberg.

We set very high goals for us when we raised our first investment in April 2013. As a startup, data is your best friend. Reviewing those goals at the end of the year, we realised that we have trailed behind on few of them. For us a team, we have always believed in chasing bigger dreams and not take up smaller challenges. Keeping up with that belief, we will be sunsetting Wishberg in coming days while we give time to our users to go through their accounts and wishes.

We're also excited to announce that the founding team at Wishberg is joining forces with FreeCharge. We believe FreeCharge will be a perfect fit for our team & capabilities - FreeCharge's scale combined with its amazing leadership, awesome team and appetite for success is inspiring for us. Pravin (@beingpractical) will lead efforts on Product & Growth and Kulin (@NowEntrepreneur) will lead efforts on Partnerships & Alliances at FreeCharge.

We can't begin to express how grateful we are for your support throughout this journey spanning over last three years. Our heartfelt thanks to our users, our team, our advisors, our investors, our early adopters, our enthusiasts, our well wishes, our friends and our family members.

- Team Wishberg

Original article here: http://blog.wishberg.com/post/92014031388/the-final-note-thanks-from-team-wishberg

Startup Failure: GreenGar Studios

Failure of a success - Let's redefine the success and failure of a startup

If success is defined by creating awesome products, we created apps that have been downloaded 15 million times across iOS, Android, and the Mac App Store.

If success is defined by people using the product, we still have over a quarter of a million users actively using our apps around the world.

If success is defined by making money for the business, we put in our bank account $1 million in total revenue.

If success is defined by having investors believe in your team, your business, and your product, we successfully raised capital from the top super angel and angel investor in Silicon Valley.

We are the first company from Vietnam that got accepted to the most prestigious accelerator program in the world.

From 90 countries around the world, our Whiteboard app won No. 1 in the Dragon's Den Pitching Competition by the World Bank's infoDev.

We could list countless press and media coverage from all around the world.

But here is what we could not do: we simply failed in creating a scalable business!

There are so many lessons learned from this journey. Five years is not an overnight success nor failure—our team has been working incredibly hard to deliver the best apps for our users with limited resources.

This blog is not an explanation but rather an appreciation to our supporters. I'm thrilled by all the support I received from everyone from Vietnam to Silicon Valley—from finances to time, especially our co-founders, teammates, advisors and investors. Our users have been the biggest inspiration for us to continue developing many awesome apps for the last five years.

If you still remember my pitch: "We already made $1 million. Let's talk about $1 billion." Maybe one day that statement will come true, and I'm still working very hard on that. However, it will not be with my GreenGar chapter.

This is not the end …

This is the new beginning for GreenGar's members .

Over the years, we grew up with all of your love and care—from many people that we met intentionally and randomly. To us, that's the bigger success of our journey!

In the last four months, our biggest challenge was to make a decision about our apps. There are hundreds of thousands of people still using them every month, including many schools from kindergarten to university. We decided to leave the apps on the App Store and Google Play for our beloved users, but there will be no further development nor support.

Thank you for your understanding. See you in my next chapter shortly!

Original article by GreenGar here: https://medium.com/@greengar/failure-of-a-success-6ce0597189e3

Startup Failure: Rivet & Sway

The inside story of Rivet & Sway: Why this online retailer closed its doors

Rivet & Sway, the online retailer that specialized in eyeglasses for women, unexpectedly closed its doors this week just 16 months after raising a $2 million financing round.

The news was greeted with sadness by many of the company's loyal customers, who purchased stylish frames on Rivet & Sway initially for $199, and then later for just $169.

Rivet & Sway tried to lure female shoppers with stylish frames and seemed to have a lot going its way: a growing customer base; attractive pricing and a well-regarded CEO in former Amazon.com manager Sarah Bryar.

So, what happened?

Bryar shared some of the pitfalls with GeekWire, providing a level of transparency not often seen with those dealing with the pain of closing down their entrepreneurial dreams.

Rivet & Sway — which raised more than $3 million in funding — did face serious competition from heavily-funded Warby Parker. But Bryar said the 14-person company's challenges did not stem from that rivalry. In fact, she said Warby Parker helped get more women comfortable with the idea of buying eyeglasses online.

Rivet & Sway's problems really were tied to the high-cost of customer acquisition, especially around the costs associated with a service that allowed women to try on frames at home.

"Women want to try frames on before purchasing, but it's an expensive marketing program to ship frames back and forth," explained Bryar. "Our economics were improving, and we had plans to introduce hyper-efficient showrooms to attract customers not yet shopping online, but our progress wasn't strong enough to attract additional capital. We had an option on the table to strip back our team and value proposition, but I felt it would compromise our ability to differentiate on service and quality. And so, we made the hard decision to close the company on a high note."

Here's more from an e-mail Q&A we conducted with Bryar Thursday:

What was the biggest surprise — something didn't foresee happening?

"I'm not sure I can call out a single "biggest surprise." When you're forging a new path, there are always things that are surprising along the way because you don't know what to expect. Hence the fun of a start up!

One insight I found fascinating: women are truly motivated by getting a great deal. We did a lot of experimentation with promotions to influence conversion. On one hand, common wisdom is if you're going for a brand play, don't discount. Don't train your customers to wait for the sale; invest in the brand to justify a higher price point.

On the other hand, discounting truly works — every time. We changed our pricing in January 2014 to maintain the "premium" price anchor at $169 that signaled quality, but we also provided a motivating incentive to buy additional pairs at $99. It was really working well. By the way, I put "premium" in quotes because $169 is a premium to Warby and other Warby copycats but not compared to pricing in legacy channels where $169 still offers a disruptive value."

Do your challenges speak to the difficulty of selling something online that people are used to purchasing in-person?

"Yes and no. 96 percent of all eyewear purchases still happen in traditional retail channels. But because the process quite frankly sucks — expensive, time consuming, uninspiring — we found that women were quite willing to give us a try especially because we addressed these pain points. However, driving awareness among potential users is much harder because online penetration of eyewear is still low. So when you're an e-commerce company with limited funds — in contrast to Warby — relying on word of mouth and online search, it's just harder to reach those potential customers."

If you could do it again, would there be one thing you'd change with how the business was run?

Early members of the Rivet & Sway team Early members of the Rivet & Sway team "If only running a business were

straightforward enough to boil down to one thing I would change to effect a different outcome! Hindsight is 20/20 (no glasses needed), so there are a lot of things I would do differently.

Here are a few at the top of my list:

Drive market awareness/dominance regionally: Seattle, NW, West Coast… Adapt the Home Try On model sooner Scale only when absolutely necessary (we outsourced to a big-company 3PL way too early) Focus more on PR Get more sleep

Aspects I wouldn't change: Our authentic commitment to delight our customers High standards for design: of our collection, the website, packaging, emails, etc. Our target audience Our team

On what worked, and the overall challenges of the business model:

We designed and manufactured our own frames so that we could offer glasses at a disruptive price point. We helped women find stylish specs via our Personal Stylist and Style Finder services, both of which delivered high conversion and customer satisfaction.Our pricing model, $169 + $99 each additional pair, influenced women to buy more than one pair in every order.

Women loved Rivet & Sway, consistently rating us off the charts for customer service, and referring us to friends. 75 percent of our visits came from referrals, organic search, direct visits, and content marketing like social and email, i.e. non-paid channels. And as I mentioned before, our Net Promoter Score was consistently between 90-95.

The part of the model that was challenging was the physical home try-on process. Even if you can reduce the cost of the back and forth shipping and refurbishing trial frames with scale, the conversion rate has to be high, north of 40 percent, for the cost to justify itself. Let's say it only costs $15 per home try on. At a 40 percent conversion rate, the cost per converted order is $37.50 — and that doesn't include the marketing cost to acquire the customer in the first place.

Warby Parker also offers home try on, and there has been speculation that one of the reasons they're opening up retail stores is because home try on is very hard to scale. Despite advances in virtual try on, customers — particularly women — want to try on frames before they purchase. Maybe that will change over time, but we're still very early in the adoption curve so physical try on is still expected."

GeekWire's Taylor Soper contributed to this report.

Original article here: http://www.geekwire.com/2014/inside-story-rivet-sway-online-retailer-closed-doors/

Startup Failure: Dijiwan

Why our startup failed

One would never expect their startup to fail 6 months after having raised half a million euros from public investors. It eventually happened to me, as the CTO of Dijiwan, a former digital marketing startup based in Bordeaux, France.

tl;dr

A good product idea and a strong technical team are not a guarantee of a sustainable business.

One should not ignore the business process and issues of a company because it is not their job. It can eventually deprive them from any future in that company.

Why writing about it only now?

Success and failure are bound to Time. And Time flows differently whereas you are in a rising or an abyssal phase.

We initiated the Dijiwan project around September 2010; We initiated the Dijiwan startup around September 2011; Dijiwan raised 0.5M€ in January 2012 from French public investors to develop its business within a year; Dijiwan started struggling to clear its paychecks in July 2012; Dijiwan requested bankruptcy protection in September 2012. Why write this side of the story a year after the bankruptcy? Why not sooner?

Now is a good moment. Prior to now, I would have put too much emotion into the analysis. It could have been perceived as some sort of revenge or an easy backstab of a rotting corpse.

One sometimes needs time to step back and to acknowledge one's own failures.

Trust your gut feeling

I remember my first conversation with the CEO of the company: it went directly to the bin of my memory. Too many buzzwords. Too much flattery. So nothing much, really.

I learned about that the second time we met, as I did not recognise him while he did recognise me.

The dating continued a couple of times more: I was definitely attracted by the innovation of the product (mining and mapping networks of Web content). I was not convinced by the CEO… but well, as the idea was to raise money first, I decided to ignore my gut feeling and to aim for the success of this operation.

Raising the money equaled building a dream team in ideal conditions. I could not believe it could not work. So I purposely ignored my initial judgement.

Terrible hiring and management

Each hire should be tight to objectives, costs and metrics.

- Is the hire generating revenues, is it creating product value or is it sustaining the organisational growth?
- How do I know if the job is properly done? Have we agreed on the contract termination conditions?
- Is the hire the result of an internal inefficiency or * is it required due to work overload?

We had 2 marketing officers, 2 business developers, 2 strategy analysts, 1 researcher contractor and 4 developers when operating at full speed.

If we analyse the figures:

- 2 people had the responsibility to bring money on board;
- 2 people had the responsibility to broaden our visibility;

- 3 people had the responsibility to deliver customer studies and analysis (1 analyst + 1 researcher);
- 5 people had the responsibility to build the innovation (4 devs and 1 researcher).

In addition to the previous numbers, we fired 2 marketing officer, 1 developer and 2 design contractors.

This is both too much and not enough at the same time:

- we signed less than 100K€ of contracts with 3 or 4 customers;

- we had too many people to produce customers' deliverables;

- the high marketing turnovers reveals a large bottleneck and inefficiencies.

One of the failures was clearly the will to spend all the money rather than to spend it wisely. The primary objective should have been to generate more revenues or to sustain the innovation — if it broadens the business reach and simplifies customers acquisition.

Reckless expenses

Around 600K€ were burnt in 6 months if you consider the revenues and the funding. For a 10 people company, this is a lot!

Barcelona's Mobile World Congress 2012:

- I remember seing nice pictures of nice cocktails and nice restaurants from the business developers;
- I remember seing lots of appointments with potential customers;
- I remember it ended up with no customer but a partnership;
- I still do not know the cost of the operation, even as a cofounder.

A customer trip's cost to visit Lisbon:

- 6 people trip with paid hotel and so on;
- 1 person speaking to showcase our work and our product;
- The goal was to sign new contracts with other affiliates of the group;
- No contract has been signed therefore;
- Again, the cost of the operation has never been communicated.

I never had access to the accountancy results. It bothered me but I did nothing because I never thought it could possibly be that bad, that fast.

The cherry on top: we received a massive LED TV screen to demo the product to customers in the same month the first paycheck was rejected.

Unbalanced partnerships/contracts ratio

A fresh and innovative company can struggle to convince people to pay for their service even though they have no public credibility yet.

We granted a lot of discounts to get customers on board. For the sake of signing new and more cyclic contracts afterwards. The contracts never happened.

We most often ended up accepting a partnership. Which means producing a free work as a proof of concept that our product worked for the customers. It was associated to a shallow promise to sign regular and paid contract afterwards. It never went further the free work as a lot of excuses suddenly appeared: we will see later, our marketing manager got a promotion etc.

Providing free work has a cost.

A 2 week free work effort costs a month of salary of every person working in the company. So it is certainly not free.

Waiting for customers, waiting for customers to make a decision, hoping a customer will say yes, waiting for the customer to sign the check after saying yes is a very long process. During this time, your company has to pay for expenses, costs and salaries.

Work is not free. Not signing a single paid contract is called bleeding to death. And this death is a slow process.

Big customers = lots of time

Speaking of customers, the main lesson here: the bigger the customer is, the longer the process is.

A small company has time to choke several times during a single big company heartbeat. A small company has time to die several times during a single contract decision process of a big company.

The promise is the following: a startup thinks it has an innovative solution. A startup thinks many other companies would pay for it.

However these companies made their life without you, most probably for years. In the end, they do not care much about you. They will listen to you. It will be entertaining. It will create hope. But they want to make sure you will exist in a year or two, at least.

They will most likely not eat out of your hand if they can perceive you need them at first. It is like a hobo trying to date the Prime Minister because he found the secret of political success. Even if the promise is beautiful, it will not happen. Never.

An innovative company needs to convince first. It needs to showcase its product, to demonstrate its benefits and the value added to any customer willing to pay for it.

In order to summarise the startup paradox:

- a startup does not have time and needs money to survive;
- your potential customers have the money and need time to see if you will survive;
- the money you do not have yet is money you do not have at all.

Dijiwan announced its 1M€ contracts objective for December 2012 whereas it actually signed contracts for less than 100K€ in total (article in French). It was in May 2012. The first choke happened only 2 months after.

The bigger the lie is, the more inclined to believe it you will be. Except your potential customers. They will wait. Their timescale is not your timescale.

The CEO bottleneck

Our CEO had charisma. A lot of charisma.

I reckon if I did not trust him at first, I trusted he could manage to get the business to rise and to shine. As his words were strong and beautiful, I ended up trusting him.

Trusting is accepting. Successes and mistakes. And one needs time to acknowledge their trust to be a mistake.

A lot of decisions appeared to be wrong in the end, mostly thanks and because to his charisma and its relationship with our weaknesses:

- the hires depended on him (except for the tech team);
- the contractors depended on him (even if no one agreed);
- the accountancy depended on him;
- the business depended on him;
- the product development did not depend on him.

Can you see the bottleneck here?

The decision process was top to bottom by authority. The only area out of his scope was the innovation and the technical team.

The CEO of the company did eventually know how to manage a company like a boss. But he did not know about his own product. Therefore he was neither convincing nor able to sell it. He was selling carpets rather than the crafted outcome of his soul.

Because of that disconnect.

- A startup does not need a CEO. A startup does need a steward.
- A startup does truly need a leader rather than a director.
- A startup does need champions first.

This is the credibility of the company: its people, their individuality and their experiences.

Ignoring problems

I mainly ignored problems because I never thought how bad it could end up with. I thought we would shut down the company if we had no more money to pay ourselves. I thought we would know if we did not have money anymore.

Many symptoms accumulated along the weeks:

- no contracts signed despite the numerous business trips to Paris;
- endless negotiations with big customers;
- endless negotiations for a next funding round with private investors;
- 6 months for the marketing team to produce a 6 pages website;
- no follow up from hot customers from a month to another one;
- more and more promises during our monthly board meetings;
- extreme difficulty to find out precise and quoted details about expenses, accountancy and so on.

The tension between us (the cofounders) eventually arose and grew. I paid less attention to ours conflicts because I felt it was useless to waste energy. I rather preferred to focus more on the doing part of the product, to care even more of the people. Not to buy a conscience, but because I felt wasting energy in verbose and sterile conflicts would be no help.

Later is not an answer. Promising millions is never an answer either. Neither is ignoring the problems, and thus my responsibilities.

You need to add value, not complexity

I thought the product we made was worth it for the customers we were targeting: marketing companies, companies in a need for an online strategy and companies in a need of improving/measuring their online strategy.

We created the company on top of an existing proof of concept of online data mapping, entirely manual, long and error prone. As the idea was to automate and hasten that process, I never asked myself the question if it was that the customers needed.

I think we had too much hesitation in the decisions about what we were able to do and on the delivered product as well.

I mean, who in the marketing industry is able to decode a 15K€ Web graph even if it contains loads of useful informations? And that was our number #1 feature.

Now, I acknowledge the graph data was essential. We should have delivered them in an more effective and practical way: spreadsheets, data views (who are the influencers, how can I contact the websites owners, recommended keywords for Facebook/Twitter/newsletters per group of users etc.).

We should have sold a set of of lists with pictures, a few simplified maps and targeted howtos rather than an army-grade Web satellite radar with full control in your hands.

The learning curve of providing useful and high value data is rather low compared to putting all the informations in a powerful and new, and thus complicated tool.

Most of all: it would have been easier to understand. Thus to sell, to grow and to convince. And for us to survive.

We built a nice piece of software. But we never acquired any additional customers by polishing a line of code or by adding a new feature. Because those lines of code never faced anyone willing to sign a check.

You do not need money

A private investor once said to us:

I want to make sure you do not need my money. It might be strange coming from an investor but I reckon it is a smart advice. Investors should either help in either ways:

- idea bootstrapping through a cheap prototype to test a market;
- prototype iterations to test even more a market;
- business growth based on a proven (yet successful prototyped) idea.

No one will blindly throw a million or two in your company because you have a cool idea. It will happen only if you have a good and proven track record, a renown team, a low risk/high return on investment business or a high risk/flourishing activity.

At the time, we needed money to survive. I did not know it. But that was the case. The investors wanted to see if our idea and working MVP was capable of succeeding in the real world. The world where other companies buy your product.

Also, customers will not sign a check to a person they do not trust. I discovered that way too late, when the company collapsed: former prospects acknowledged they did not trust our CEO, even if they loved our idea and product. They smelled something wrong.

Damn you bottleneck!

Pulling the fire alarm

When do you need to pull the fire alarm? What to do if the company goes down into madness?

Surprisingly, a company officially goes wrong only when it can be proven. Which in financial words means: when a company has debts.

As a cofounder, I should have acted quicker. As soon as the first employee notified us about his rejected check. As a bank never blocks an account straight off, you can imagine it has been going on for a while.

What could I have done? Ask the CEO to leave? My only evidence is that we had run out of money. I still wonder if debunking a problem from the inside is doable on your own, without the consent of a majority.

Instead of acting, we had continued trusting the CEO despire a lot of suspicion. A very largely tainted trust I admit.

We eventually waited 2 months before declaring unpaid salaries to the Chamber of Commerce and to the local Job Office.

We got a little help from trade unions, and mostly figured out how complicated our situation was: a locked cage above a precipice.

When the CEO learned about our action, he anticipated the move smartly: the Chamber of Commerce canceled our audience because… the CEO has requested this very same Chamber's protection!

A way to hold bankruptcy by freezing the debts while staying on board. At the time, the ambiance in the office was nasty and no one had any motivation neither trust regarding the CEO.

We had to push hard to get fired. Yep, because everybody wanted to leave. By being fired, people could at least sustain temporarily on government benefits.

When you have a continuing contract in France, you are bound to it. If you want to break it, you can. However you will not get any government benefits while you search for a new job; even if it is due to a lack of responsibility of the employer.

If you want your money back? You need to hire a lawyer to go to a special court. Which means committing money to get your money back. A year later.

This is that long of a timescale.

Where are the public investors?

We also pulled the fire alarm with our public investors. We had an emergency meeting. A meeting where the CEO promised 3M€ of potential contracts. I can still giggle nervously when I think about it.

We had to send them a report. It was sent. Thus I do not know about its content as at that time, I was already pulled off the communication channels.

Since then? Well, not much. Apparently saving a half million of public investment is not worth it. Not we were requesting money, but just help. To clean up the mess and to heal the pain.

I have not been aware of any feedback after we got the money from funding. To verify everything was okay or even to connect us to other useful persons to fuel the growth.

NO-THING.

They only cared about us creating jobs in the region. When it was about unpaid employees suffering they did not give a shit. And for that, I am still very angry about this irresponsible behaviour. This experience gave me a dirty after taste of how money can be more destructive than constructive.

Conclusion

Some of the mentioned problems are not a potential sign of death on their own. You can ignore problems for sure. You can escape from your responsibilities for sure. But the job needs to be done.

If the job is not done, that is the problem. Period. I now think ignoring that problem is a fatal mistake. And we paid the price for it.

Startups are not bad. Public investment is not bad. Slow justice is not bad either.

What is wrong is how freaking difficult it is to extract yourself from this mud, willing and helping your pairs not to drawn in something you created.

What is wrong is that fucked up money thing to make more money for the sake of making more money. Or to ignore the loss of half a million because you know, it is better that a local innovative company dies rather than acknowledging it should have required a little care.

If I had to do it again? I would do it again, differently. Without the scale in mind. Without the wrong people. For a right balance.

Original ariticle here: https://oncletom.io/2014/why-our-startup-failed/

Startup Failure: Wantful

Some news about Wantful

I'm profoundly disappointed to announce that Wantful has suspended operations.

We accomplished a great many things in our 18 months in market: an enormously well-received gift offering; widely-lauded experiences for web, tablet, print, and in-store; a network of 600 vendors creating the most exceptional products around, with a robust and streamlined logistics architecture to support them; an assortment and an approach to content that inspired and engaged; and the brilliant and tireless team responsible for it all.

What we did not accomplish yet is the kind of highly accelerated growth required to secure later-stage venture capital, despite the enduring enthusiasm around what we've built.

The coming holiday season was shaping up to be pivotal for us, but the loss last week of a planned follow-on investment from a strategic partner leaves us little time to secure an alternate source of capital, or to pursue the other opportunities on the table.

Our top priority right now is exploring all our options in an orderly fashion—and in taking care of our customers and the team—but if I can be of assistance please reach out to me personally.

Whatever the outcome, we're deeply grateful for the support and encouragement we've received from our customers, friends, investors, and partners through these past two years, and honored to have had the opportunity.

Original article here: http://johnpoisson.com/post/60467938116/some-news-about-wantful

Startup Failure: Disruptive Media

The Disruptive Adventure 3: One Adventure Leads to Another!

This is the final part of The Disruptive Adventure, where we see that the whole adventure led to something good, even though it didn't go as initially planned. This post makes more sense if you have read part1 and part2 first.

From the beginning we had all the time planned to sell photo books in PiX'n'PaLs. At this point we had built the software needed to setup a store selling customized photo products. Additionally, we had sold the software to customers, who had all the capabilities to print photo books and other products. Customers of StoreFront were more than happy to handle our production. This was also technically easy for us. All we had to do was transfer our own orders to our production partner's account. We could charge our end users whatever we wanted for their products and then pay the production partner an agreed amount.

We thought that others would also be interested in such a feature for their website, that made it possible to sell print products from all their pictures. Therefore, we decided to build something more generic, that everyone else could use as well. The new service was called PiX'n'PaLs PhotoProducts, which helped publishers monetize their website by selling print products from images. We took care of payment, shipping and everything else. The publisher got paid a 20% commission.

The service had two main features:

A publisher could add our code snippet to their website, making it possible for users to buy print products directly from an image. Clicking on the buy link opened a small shop, where the user could select which photo product to order from the image and use basic image editing tools, such as cropping, rotating and resizing the image.

Integrate our whole e-commerce solution to their website. When a user ordered print products, they had all the photos immediately available. This was practical in PiX'n'PaLs, where users had access to all their pictures and their friends' pictures from the shared events. The events appeared as directories in the shop.

We launched with one of the biggest Finnish websites, Riemurasia, which enabled this feature for all its photos. The launch looked really promising and we immediately got lots of orders. Unfortunately our luck ended pretty soon when our feature had to be limited to only certain photos, due to potential copyright issues. Later it also turned out to be so that many orders got canceled, so in the long run it wasn't that easy money after all. We would have needed many more big websites and better products with higher margins. Finland was definitely too small.

Launching this globally would have required lots of funding in order to get production and logistics to work well internationally. Getting deals with big international companies was hard and plugging into their production pipeline turned out to be technically impossible, since they did not have any APIs. It's hard to tell whether this would have worked, since we were running out of money and had to leave it there. Potential investors were not too crazy about investing in a declining market either. The numbers did not fully work out.

The safe way to fail for many entrepreneurs is to start doing consultancy. This is what we did. We started doing projects for others to get some cash quickly. This seemed more comfortable after a long adventure with very high risks, so we kept doing it for full time until we joined one of our customers (Kiosked) as partners. We had already built the first version of their software and had all knowledge to take it forward from there. Today I am working there as the Chief Technology Officer and living really exciting times now that the company is growing rapidly.

PiX'n'PaLs is still running and used regularly by a lot of users and PiX'n'PaLs StoreFront is still used by two print-labs. These products are not developed further anymore. Some maintenance work is done every now and then to keep paying customers happy. If we would want to continue one day, we would have to update the UI and graphics completely, since they were optimized for IE6 and other ancient browsers. We have shut down the PhotoProducts and our own store (EasyKuva) which was one service built on the PhotoProducts API.

Well, you probably wonder: was it worth it? Absolutely! I don't regret anything. The things I experienced and learned could not have come from any normal nine-to-five job. And as it turned out to be, one adventure led to another.

Startup Failure: Calxeda

Low power WON'T bag ARM the server crown. So here's how to upset Intel

Exclusive In the last days of 2013, Calxeda, the ambitious startup that hoped to design ARM processors for data-center servers, imploded.

Now El Reg has sifted its ashes, and pulled out some advice for the silicon upstart's contemporaries.

The demise of Calxeda caused many to ponder the viability of general-purpose ARM-powered computing in the data center, especially given Intel's twin threats of its custom chip business and a newfound dedication to shrinking the power consumption of its Avoton server chips.

Calxeda's plan was to design chip packages that combined its network fabric electronics and other controllers with processor cores licensed from ARM; the resulting blueprints would turned into working silicon by outside fabs, and used in kit from the likes of HP. The startup's low-power ECX-1000 used the quad-core 32-bit Cortex-A9 architecture; the ECX-2000 used the Cortex-A15.

But 64-bit parts, desirable for enterprise-grade workloads, weren't scheduled for release until 2014, a date the company just couldn't reach before the lights were switched off.

Last night, El Reg approached Karl Freund, Calxeda's former veep of marketing, and he told us about what other ARM-powered startups can do to avoid Calxeda's fate, and why major adoption of the chips is likely, but not for the power-consumption reasons promulgated by the press.

"In high tech, we are all trained by the years of the dot-com boom to think that being first to market is critical," said Freund, who is now an independent analyst and consultant on ARM servers.

"In [Calxeda's] case, we moved faster than our customers could move. We moved with tech that wasn't really ready for them – ie, with 32-bit when they wanted 64-bit. We moved when the operating-system environment was still being fleshed out - [Ubuntu Linux maker] Canonical is all right, but where is Red Hat? We were too early."

Calxeda's folly, Freund reckoned, was that it came out early enough to ignite a huge amount of interest from the industry and coverage in the media, but was too early to satisfy key customers with serious spending power.

Though large-scale operators such as Google and Facebook are all interested in ARM-driven servers – a senior Facebook executive joined Calxeda's board in October – they are all going to wait until 64-bit ARMv8 packages are available, Freund reckoned, and will likely want some kind of software ecosystem to be present as well.

Chips stuffed with 64-bit ARM cores are expected to come along in the first months of this year, though Freund believes it will be in 2015 that the industry adopts the architecture in a big way.

In the year between release and adoption, in a sense we'll witness a Mexican standoff: the IT world's superstars won't run production systems on computers until they're demonstrated reliably running critical software. Someone will have to crack and take a shot on the platform before others invest time and money in testing and validate their applications for wide ARMv8 deployment.

"The big guys, they'll all experiment with ARM, many of them will deploy ARM, but not until first-generation 64-bit at best," Freund said.

"These guys have advanced technology strategists so they're always looking at what's next on the horizon. We have had deep, deep discussions with all of them about our technology and about our roadmap. They're all going to wait for 64-bit.

"Red Hat will not have [an ARM] RHEL until 2015 at the earliest. If you're an Amazon you're not going to stand up a big farm of ARM servers if there's nobody running a big load of software on ARM."

The reason ARM will be successful is not performance per watt

ARM-compatible processors tend to end up in handheld devices, and thus aren't as battery hungry as their x86 cousins. In these eco-friendly times, the suggestion you could run an ARM server farm off, say, a solar panel array was enticing for companies with hefty electricity bills.

If power-sipping 64-bit ARMv8 server chips have a chance of serious adoption in the data center come 2015, there's a good chance Intel will at that time be fielding some capable Atom processors with close-to-ARM milliwatt-per-MHz ratings thanks to Chipzilla's advanced manufacturing expertise.

Freund says this is "a very cogent argument," and claimed somewhat heretically that "the reason ARM will be successful is not lower performance per watt."

A minor performance-per-watt advantage (and by 2015 all signs point to it being minor at best) is not great enough to get people to switch architectures, he said.

Instead, the flexibility of the ARM architecture will be vital – specifically, the freedom to license a streamlined ARM core design and bolt on whatever custom hardware is needed to perform or accelerate a particular task. This approach contrasts Intel's beefy but general-purpose packages.

Just as smartphone makers turn to system-on-chips with touchscreen-driving GPUs and wireless networking built alongside battery-friendly processor cores, server manufacturers should be able to pick up parts that strongly glue multiple cores to gigabit ethernet and high-end SATA controllers.

"It's the ability to customize around ARM's intellectual property, which you cannot do with Intel intellectual property," Freund told us. "If you're Google looking at some kind of acceleration, or Sandia Labs looking at very specific algorithms you want to tune for, or you like the [Calxeda] fabric and want disaggregated resource models – that's why you pick ARM."

This lines up with El Reg's own analysis in December, and the thoughts of a well-placed source within the semiconductor industry.

Though Intel has an early-stage custom x86 chip business that already works with eBay and Facebook, Freund believes the pace at which a company can either buy and modify ARM chips, or contract to a third party like a Phoenix-rebirthed Calxeda to do so, will beat Intel every time.

"Either customize for a very large account or customize for a very specific workload or market segment," he said. "My advice to other ARM vendors - I wish them all the luck - is you either have to add significant added-value or very, very low costs."

Pay attention, Applied Micro's "X-Gene" team, and AMD's "Seattle" crew, or else you risk a dim reception to your promises of future chippery. ®

Bootnote

Intel experimented with ARM-compatible system-on-chips called the Xscale family between 2002 and 2006, at which point it offloaded the tech to Marvell. Chipzilla pitched the processors at mobile phones, networking kit, disk controllers and similar embedded computing products. Calxeda clearly thought bolting cores onto specialized hardware was the way to go: its co-founder Barry Evans once ran Intel's Xscale businesses.

Original article here: http://www.theregister.co.uk/2014/01/08/calxeda_postmortem/

Startup Failure: Turntable.fm

Shutting down

After four years, we shut down our final product, Turntable Live, in early January. A lot happened over those four years. We built four separate products (stickybits, turntable.fm, Piki, Turntable Live) and I had the privilege of working with so many amazingly talented people. I'm incredibly proud of the products we built together and thankful for the new friendships created.

I used to say the hardest part of running a startup is hiring. I learned that shutting down is harder. From an outsider's perspective, it's a light switch. It used to be on and now it's off. But the wind-down of a company takes time and it's something every startup founder fears could happen.

You see the end on the horizon and work as hard as you can to push further away from it. As it gets closer and you realize there's no other option, you begin to figure out the logistics of shutting down. Layoffs are the most painful part.

You then have all this stuff. Laptops, TVs, monitors, dishes, a drink refrigerator, tables, desks, chairs, a lease, and tons of monthly services. Slowly over several weeks, you work to get rid of all of it. And it's heartbreaking to see it all get slowly torn apart.

Shutting down has shown me the ugly side of startups. It's a side where some investors break all communication or turn against you. It's a side where close allies turn away. It's a side where all the promises you made to your team go unfulfilled.

All of this is chock full of important lessons. It shows you how the people around you handle both the good and the bad. You see the people who will stick by you, believe in you, and are ready for round 2 (or 3 or 4).

Turntable.fm grew into a wonderful community that spawned thousands of friendships and even a few marriages (including an employee to someone he met on turntable). We employed a team that varied in size from 2 to 20 for four years. We created a type of music sharing that still exists in products inspired by us.

Ultimately, I didn't heed the lessons of so many failed music startups. It's an incredibly expensive venture to pursue and a hard industry to work with. We spent more than a quarter of our cash on lawyers, royalties and services related to supporting music. It's restrictive. We had to shut down our growth because we couldn't launch internationally. It's a long road. It took years to get label deals in place and it also took months of engineering time to properly support them (time which could have been spent on product).

When we were first launching our servers, we needed a good naming convention for them. I jokingly thought of naming them after failed music startups. At first it seemed cruel, but it grew on me because I wanted it to remind us of the lessons from the past and also thought maybe it would be good luck (like telling an actor to go break a leg). The joke wasn't lost on us when we named our final server 'turntablefm.turntable.fm'.

The key to all of this is to keep building. No one has any idea what is going to work and what's not. Don't listen to the people who think they know. Sure, this one didn't pan out, but each failure helps us navigate the thousands of decisions we will need to make for the next one. That knowledge helps us build better things that will last longer.

And that's what I will continue to do. It's customary for people to take a couple months off in this position. I decided instead to jump right back in. I've been working with a couple people and started a new company.

We were able to quickly build our first app. It's called Ketchup. But I don't want this post to be about that, so you can read about Ketchup here if you're curious.

Original article here: https://medium.com/@billychasen/shutting-down-7958dae1d27b

Startup Failure: Tutorspree

When SEO Fails: Single Channel Dependency and the End of Tutorspree

Although we achieved a lot with Tutorspree, we failed to create a scalable business. I've been working through why. In doing so, I'm trying to avoid the sort of hugely broad pronouncements I often see creep into post mortems that I've read: "don't hire people!"; "hire people faster!"; "focus on marketing at all costs"; "ignore marketing, focus on product" etc.

I've focused here on the strategic causes of our failure. While I learned a huge amount about operations, managing, and team building; mistakes made in those areas were not the ultimate cause of failure just as the many things we got right within the company did not ultimately lead to success. I also recognize that this doesn't cover every detail, even on the strategy side.

SEO: Too good to be true

Tutorspree didn't scale because we were single channel dependent and that channel shifted on us radically and suddenly. SEO was baked into our model from the start, and it became increasingly important to the business as we grew and evolved. In our early days, and during Y Combinator, we didn't have money to spend on acquisition. SEO was free so we focused on it and got good at it.

That worked brilliantly for us. We acquired users for practically nothing by using the content and site structure generated as a byproduct of our tutor acquisition. However, that success was also a trap. It convinced us that there had to be another channel that would perform for us at the level of SEO.

In our first year, that conviction drove our experiments with a series of other channels: PPC, partnerships, deals, guerilla type tactics, targeted mailings, craigslist posting tools, etc. Each experiment produced results inferior to those from SEO. The acquisition costs through those channels were significantly higher than what was allowable based on our revenue per customers. We also found that potential customers coming through PPC were converted at a lower rate than those originating through SEO. Even as we sharpened our targeting, experimented with messaging, and sought advice and consulting from more experienced parties, we found that paid channels just weren't good enough to merit real focus.

That dynamic put us in a strange position. On the one hand we had a channel bringing in profitable customers. On the other hand, we did not have the budget within our model and product to push hard enough on other channels.

The AirBnb Head Fake

At the end of our first year, the divergence between our success with SEO and our failure with other channels dovetailed with a whole set of lessons we drew from analyzing user behavior on Tutorspree. We realized that there were fundamental problems with the product of Tutorspree which both prevented us from converting visitors to customers at optimal rates and from having enough capital to spend on acquiring more visitors/potential customers.

We had modeled ourselves on AirBnB, believing we were a clear parallel of their model for the tutoring market. What we were seeing in terms of user behavior, however, was fundamentally different. Parents simply didn't trust profiles and a messaging system enough to transact at the rate we needed. Our dropoff was too high, and the number of lessons being completed was too low. We realized that we were wrong in how we thought about the entire market, and radically altered our model to suit in March of 2012. Looking back, it is also apparent that we were able to ignore our error for as long as we did precisely because SEO worked as well as it did. The dynamics of our marketing provided air cover for any other issues we had.

We called the new model Agency as we pulled in aspects of a traditional agency's hands on approach and combined it with our matching system and our customer acquisition channel – SEO. Within a month of the change, we doubled revenue. Six months after the shift, our revenue had increased another 3x and we'd increased margins from 15% to 40%. The new model gave us our first profitable month, and put us within striking distance of consistent profitability. It looked like we had cracked the product problem. Our conversion rates were way up and per user revenue was climbing rapidly. Those factors gave us the budget we needed to more productively experiment with other channels.

Virtually all of our customers came from SEO.

New and Better Model; Same Old Channel

By December of 2012, we had virtually infinite runway and were at the edge of profitability. We still wanted to swing for the fences, and, given the radically shifted economics presented by our new model, we made the decision to retest all the marketing channels we had tried with our initial model and then some. We knew that only having a single scaling channel – SEO – would not let us become huge, so we began pushing for another scalable channel.

Given the strength of where we were and the challenges we saw, we raised another round with the explicit purpose of finding the right marketing channels. While we considered raising an A, we played conservatively, deciding that we wanted to find the repeatable channels, then raise an A to push them hard rather than raise too much money too early.

We finished that fundraise in January, began a much needed redesign of the site to fit with our significantly more high touch model, hired a full time growth lead and began to push rapidly into content marketing, partnerships. Then, in March of 2013, Google cut the ground out from under us and reduced our traffic by 80% overnight. Though we could not be 100% certain, the timing strongly indicated that we had been caught in the latest Panda algorithm update.

With our SEO gone, we took a hard look at our other channels. While content may have played out in the long run, and in fact showed signs of the beginning of a true audience, the runway it needed was far too long without the cushion provided by SEO. PPC - mainly through Adwords (though also through FB) – was moving in the direction of being ROI positive, but the primary issue turned out to be one of volume rather than cost. Because of our desire to focus heavily on the markets in which we had the highest tutor density (and therefore the greatest chance of filling requests), we had to carefully target our ads in terms of geography and subject. Given that dynamic, we simply couldn't find a way to generate enough leads, no matter the price. In the end, that calculus applied to nearly every paid channel we could identify.

Common Thread

Our reliance on SEO influenced nearly every decision we made with Tutorspree. At the beginning, it influenced our decisions to allow tutors to sign up anywhere, for almost any subject. On the one hand, that brought in leads we could never have specifically targeted. On the other hand, it spread our resources out across too many verticals/locations. That problem was compounded by our move into Agency. While we were converting at a higher rate and price than ever, we were also forced to spend too much time and money on completely unlikely leads. When you build your brand on incredible service, it becomes very hard to simply ignore people.

When our SEO collapsed, we routed virtually all of our technical resources to fixing it. In that effort, we had significant amounts of success. We regained most of the traffic that we lost with the algorithm switch in June. We regained a significant portion of our rankings. However, the traffic that we were getting at that point was not as high quality as that which we had been getting beforehand.[1]

Because of how successful SEO was, it was the lens through which we viewed all other marketing efforts, and masked the issues we were having in other channels along with important realities of how the tutoring market differed from how we wanted to make it work. We were, in effect, blinded by our own success in organic search. Even though we saw the blindness, we couldn't work around it.

Lessons Learned

Tutorspree taught me a lot of lessons. I learned about product, users, customers, hiring, fundraising, managing, and firing. I made some bad hires because of my own blind spots and desire to believe in how people operate. There were periods of time where I avoided conflicts within our team too much – decisions that were always the wrong ones for the business and that I regretted later. Those mistakes were not ultimately what caused our failure.

Nor is the largest lesson for me that SEO shouldn't be part of a startup's marketing kit. It should be there, but it has to be just one of many tools. SEO cannot be the only channel a company has, nor can any other single channel serve that purpose. There is a chance that a single channel can grow a company very quickly to a very large size, but the risks involved in that single channel are large and grow in tandem with the company.[2]

That's especially true when the channel is owned by a specific, profit seeking, entity. Almost inevitably, that company will move to compete with you or make what you are doing significantly more expensive, something Yelp gets at well in their 10K risks section: "We rely on traffic to our website from search engines like Google, Bing and Yahoo!, some of which offer products and services that compete directly with our solutions. If our website fails to rank prominently in unpaid search results, traffic to our website could decline and our business would be adversely affected."

For me, this is a lesson about concentration risk and control. In this case, it played out on the surface in our only truly successful marketing channel. That success wound its way through everything we did, pulling all that we did onto a single pillar that we could not control.

By necessity we had to concentrate risk on certain decisions (something likely true of most small startups). I did not have the time or resources to do everything I wanted or needed to do. I never will. But I need to be cognizant of the ways in which that concentration is influencing everything I do. I need to make sure that it doesn't dig me into holes I can't work out of on my own.

Ultimately, this post mortem is about the single largest cause I can identify of why we failed to scale Tutorspree. In examining our SEO dependence, I was surprised at how deeply it influenced so many different pieces of the company and aspects of our strategy. It powered a huge piece of our success, and ultimately triggered our failure. There's a symmetry there that I can't help but appreciate, even though I wish to hell it had been otherwise.

—

[1] This is a whole other issue I explored in the Tutorspree blog at the time. It seems that Google is increasingly favoring itself in local transactional search.

[2] RapGenius recently ran into this issue, but were able to overcome their immediate problems through some impressively fast and thorough work.

Original article here: http://www.aaronkharris.com/when-seo-fails-single-channel-dependency-and-the-end-of-tutorspree

Startup Failure: Nirvanix

A Nirvanix Post Mortem - Why There's No Replacement For Due Diligence

A couple of weeks ago the industry was stunned when cloud storage provider Nirvanix sent an email customers advising them that the company would be shutting down in a fortnight – while the consumer space is used to services coming and going, this was an unreservedly enterprise-grade service – and one that had raised $70M in venture funding to boot. Not only were customers told that the company was shutting down, but they also had to find alternative providers in very short order.

As is often the case in situations like this, there were plaintive wails from the naysayers that this is an example of why cloud computing can't be trusted and why we should all move back to the days of old. Presumably those same naysayers want us to use green-screen monitors, have a crusty old mainframe in the basement and use telex machines for communications… but I digress.

Contrary to what people might tell you, Nirvanix' admittedly awful turn of events isn't a death knell for the cloud industry, any more than PRISM is (and more about that subject another time).

What it is though, is a reminder that even with a move to the cloud and the attendant ability to focus on core business while third party minions tend to our infrastructure, there are still some requirements on customers to ensure that they, their data, and their organizations future remains secure.

Today some clarity came from Nirvanix, who tore down their website and left a statement telling customers how to get in touch with customer support. The statement helpfully advises customers that they need to actively participate in retrieving their data off Nirvanix servers and, interestingly enough, advises that Nirvanix has an IBM IBM +0.35% team ready to help customers. Many would say that this is too little, too late. That said, for Nirvanix customers, many of whom are in sheer panic about the terabytes of data they need to shift, any help is useful at this time.

I wrote a post last week setting out a couple of guidelines for organizations to avoid getting into trouble when outsourcing IT. While arguably the advice wouldn't have helped in this particular case, it's well worth revisiting and reiterative what it takes to be safe and secure when outsourcing IT to the cloud:

Do Your Homework!

The move to the cloud is an awesome thing – more flexibility, more focus, the chance to abstract non-core operations out to third parties – all the benefits we've been talking about for years. But with those very real benefits come some risks. Not risks that mean we shouldn't use cloud at all, but risks that mean that the due diligence process is more robust than ever before. Things enterprises need to think about during DD include:

- Who is this vendor? Are they well funded and secure Will their business exist tomorrow and into the future? How much downstream damage would there be if they vanished into the ether?

- What are their SLAs? Is there an escrow agreement which covers the data in the event that they cease operations? Is there an agreed export model in the event that you want out? Is there an agreement in terms of notice periods?

- Do they have pre-existing arrangements with other parties that could help in the event that they go out of business

Now in the case of Nirvanix, it's possible that DD wouldn't have uncovered issues – they were well funded, a "mature" player and had enterprise SLAs in place. But maybe their customers, given hindsight, would have been wise to negotiate hard for data escrow provisions or some other protections.

Think Hybrid!

I love the cloud – it has changed the world for the better and delivers massive opportunities for organizations. But with it comes some risks – we're increasingly prepared to trust that our data, stored in third party locations, will always be accessible. But how realistic is this? Perhaps the onus is on us as customers to ensure that there is always a "Plan B". Maybe we need to think about data storage in a hybrid way. Perhaps by using solutions that backup our data onto either

another third party or to our own local storage (and yes, I know that will sound like heresy to some).

Fundamentally, customers need to stop and think about what happens to their data when they only have one storage provider and that storage provider has a major outage, or worse, goes out of business?

Key Takeaways

The cloud is great. Outsourcing is great. Unreliable services aren't. The bottom line is that no one cares about your data more than you do – there is no replacement for a robust due diligence process and robust thought about avoiding reliance on any one vendor. Luckily there are many vendors in the marketplace who take the role of intermediaries – brokering service and storage across a number of different providers – judging by the press that one of these vendors, OxygenCloud, has been getting since the Nirvanix news hit, this is a model that more and more organizations are thinking seriously about.

Original article here: http://www.forbes.com/sites/benkepes/2013/09/28/a-nirvanix-post-mortem-why-theres-no-replacement-for-due-diligence/

Startup Failure: PostRocket

Facebook Publishing Optimizer PostRocket To Shut Down On August 15

PostRocket, a startup that helps customers optimize their Facebook posts for maximum exposure and engagement, has announced that it will be shutting down on August 15.

The company was incubated by 500 Startups, and it raised a $610,000 seed round from Polaris Ventures, 500 Startups, and various angel investors. The PostRocket team previously managed the Facebook presence of the musician Pitbull, and it dropped out of Babson College to participate in the incubator.

In an email that was sent to PostRocket customers yesterday, co-founder and CEO Tim Chae wrote: "When we first started PostRocket, we wanted to not only help marketers like you succeed in Facebook marketing, but do so with an exceptional product and service to back it. We were never able to reach the high bar we set for ourselves."

PageLever, another startup that aimed to optimize Facebook posting strategies, was acquired by social marketing company Unified earlier this year.

Chae said PostRocket customers will have until the 15th to pull their data from PostRocket, and if they're looking for another product, he recommended that they switch to Facebook's newly redesigned Page Insights.

Here's the full customer email:

This is really disappointing to say, but I must inform you that PostRocket will be shutting down its products and services next Thursday, August 15th. When we first started PostRocket, we wanted to not only help marketers like you succeed in Facebook marketing, but do so with an exceptional product and service to back it. We were never able to reach the high bar we set for ourselves. Our product had many issues and even through the down-time and bugs, you stuck with us. We thank you for that.

We should and could have done much better in bringing you a reliable product that expanded as quickly as the landscape of Facebook marketing changed. You will be refunded any of your remaining prorated credits as of today 8/6/13 and will be able to continue using PostRocket to migrate your data or find another provider until 11:59PM PST 8/14/13, at which point all data will be erased and removed.

If you're looking for an alternative service to migrate to after PostRocket, I strongly recommend using Facebook's all new native Insights product. I would have never recommended using native with FB marketing, but their new product blows any other service out the water. They are rolling out this Insights product and you should expect to receive it in the next few weeks, if you don't already have it.

Again, thank you so much for being a PostRocket customer. We really do appreciate you believing in us. We wish we could have done better for you.

On behalf of the PostRocket team,

Tim

Orignal article here: http://techcrunch.com/2013/08/07/postrocket-shuts-down/

Startup Failure: SkyRocket

A Startup Postmortem - And why vision is the achilles heel of any company

Today, I am announcing publicly for the first time that SkyRocket, the startup I've worked on since March 2013, has failed.

As you will find out if you read on, the death of SkyRocket was not as sudden as it may appear. The reality is that my company died sometime last year when I doubted myself for the first time. While I wrote about the beauty of Silicon Valley and my new life down south, my startup sat in a grave not yet marked. For about two months I was too embarrassed to admit defeat and too proud to accept my failures and move on.

In the process of starting a company I learned things about my life and the world, I worked 80 hour weeks to change, that were both painful and enlightening. This article is the first in a new series I plan to write. The theme of the series will be failure, focusing on some of the mistakes I made and how I'd advise avoiding them.

As humans, our lives are a simple game of follow the leader. During times in which we play follower, the decisions in our life are less influential than we lead ourselves into thinking. Quite often the person, or organization, that we choose to follow in the first place guides the ship leaving us to make the small choices that alter less the ends and more the means.

Recently, I was blessed with an opportunity to play the less common role of leader. I steered my own ship, and eventually convinced others to follow me. Customers, employees and mentors believed in my vision for a better world and together we slowly chipped away at a dent I promised them we could make in the universe.

In the early days of my startup, vision was trivial. It fluctuated between ambitious mission statements filled with words and phrases like "disrupt," "forever alter" and "change," and the simple goal of surviving one more day. For about a month vision was as important as my daily to-do list or a New Years Resolution, because failure affected no one save myself.

About a month after I decided to embark on my journey, I succeeded in selling my first 700 seats in enterprise software to a large university. Two weeks later I sold another 500. At the time I was selling 100% vision. After all, the extent of my software was a PowerPoint presentation demonstrating what SkyRocket could be. Slowly as the gravity of my company vision grew, so too did a weight on my shoulders. Tens of thousands of dollars had been bet and I was the favored horse in a race I hadn't ever run.

By the time I hired my first three employees, the weight on my shoulders was there everyday. In every team meeting, and phone call with a customer I was reminded of the future I promised. For months I went on believing it, myself. When the checks were coming in it was easy because my vision was constantly validated by dollar signs. But one day something changed.

Money stopped coming in the door.

In addition to a lag in sales, new product challenges arose and pretty soon I began to question myself. With each pitch following that period of doubt—whether it was to a girl at a party or an interested investor—my enthusiasm and perceived confidence dwindled.

In the months that followed I experienced vision's kryptonite in the form of self-doubt. Like Superman, I bruised easily once my incredible abilities vanished. For almost six months I had been riding on an ecstasy that woke me up at 7am everyday and kept me up until well past midnight. But in the absence of my drug—vision— I was as weak as a heroin-addict in search of his next hit.

Soon my company, customers and team became the victims of my lack of vision. And I grew fearful. Fearful that the facade I had created would disappear and fearful that those around me would notice this lack of confidence. But I'm a millennial and so in the face of failure I lifted my head higher and exaggerated daily successes more frequently.

Then I fled town.

In what would soon become the nail in my startup's coffin, I moved to Silicon Valley and left my team and customers to dry in Vancouver. I broke a Captain's one promise to his crew and I jumped ship. All the while my leave of absence was

destroying hundreds of thousands of dollars in value.

After about a month of convincing myself that everything was ok, I received a phone call from my two remaining employees.

"We're quitting and taking the consulting contract" was the long and short of the conversation.

Minutes after hanging up the phone, I realized that my house of cards had finally collapsed. I broke down and cried, knowing that everything I had worked so hard to build was in wreckage. The $14,000/mo consulting contract my team took with them didn't phase me. The sudden realization that I had failed to lead or even sense an impending mutiny crushed me and left a pain deep in my heart that will remain with me until the day I die.

In reflecting on my failure as a visionary, I am reminded of a vacation I took to the mountains when I was a boy.

My brother and I had been watching a mountain biking event at Vail for a few hours when we came across a miniature course. The sign informed us that we were allowed to give it a try.

Only 8 years old at the time, I found the cliff on the course too intimidating but my brother decided to have a go. After convincing my mom he put on his helmet and started down the skinny trail. As I watched him ride, I was inspired by his ability to look ahead rather than focus on the stream 6 feet below him. For the first minute he seemed unphased by the drop but then something changed. A missed gear caused him to readjust and then look down.

Seconds later he was laying on his back, bike on top of him. Had my brother pedaled confidently and continued to look ahead he could have avoided the fall, but instead hesitation had him bloodied and bruised in the flow of a stream rather than gloating at the finish line.

This all happened, more or less, within 9 months. Compared to successful entrepreneurs my time in the leadership seat was a mere flash, but to me it feels like a lifetime. I learned a lifetime of lessons in leadership, business and happiness. I met a lifetime of incredible people. And I did what everyone should do once in their lifetime—I followed my dreams.

My bruises will heal and I'll find a way to make my money back. I'll learn to get back on a bike and one day I'll jump right back in the leadership seat, bet it all and convince another group of individuals that we can make a dent in the universe.

Original article here: https://medium.com/@curious_founder/a-startup-postmortem-4a51006f19a

Startup Failure: GameLayers

A Story of GameLayers, Inc.

Making online social games 2007-2009 by Justin Hall

Between 2007 and 2009 GameLayers made a multiplayer game across all the content of the internet. This Story of GameLayers covers prototyping, fund raising, company building, strategic shifting, winding down and moving on.

Introduction

New communications technologies create new opportunities for people to work, play and live together. In the late 20th and early 21st century, humans created businesses on top of rapidly evolving communications networks. This is the story of an internet startup - starting an entertainment software company to evolve this new medium for human expression.

Starting in 1994, I worked for a series of internet startups based in San Francisco. Hot Wired, electric minds, Gamers.com - each of them filled an empty space in our internet and offered the public free access. Each of them received far too few dollars in return to continue.

Then in 2007 I had the chance to establish and run an internet startup as CEO. At my company GameLayers we worked to transform web surfing into a giant ongoing social game.

It was a bold adventure, and by the end of 2009 GameLayers too had failed as a company. The games we made failed to find sustainable audiences. No one works for GameLayers now or plays GameLayers games.

You are holding a story of GameLayers. I am telling this story because it helps me feel that GameLayers was a worthwhile experience. This is the story of sharing an idea. Finding money and comrades to transform that idea into a business. Working to make that business survive. Then grieving and learning when the business fails.

Along the way I had to do many things I'd never done before, things I have recounted here. I had to hire people, fire people, convince people to give me money and convince myself I was qualified to honor 120,000 users, nine GameLayerers, four investors and my marriage partner co-creating an usual game.

This GameLayers story is supported by a wide range of documents. Herein you'll find links to pitch decks, the story we told investors to raise money. You'll find a link to the "term sheet" from our investors listing the conditions for 1.5 million investment dollars. You'll find the board meeting slideshows accounting for our progress towards our goals. There's even more supporting material linked throughough: I erred on the side of too much information. You can decide how many of these linked documents you care for; hopefully these other documents are viewable on the device you're using to view this document.

My goal is to help someone with an idea see how they might possibly start a business. Maybe you don't expect to run your own company (I didn't either). Maybe you're just curious about the "internet startup" hoopla, or you dream of inventing your online game idea some day. I hope this story of GameLayers might help you better understand what can happen when people have a chance to turn an idea into a company.

2006 Grad School Project

In 2006, I was an MFA graduate student at the University of Southern California Film School Interactive Media Division.

A reformed journalist, I attended grad school to learn to make the kinds of software I'd only written about before. I was watching people play World of Warcraft and I wanted to have that kind of immersive persistent social travels through dataspace. WoW looked like fun! But I couldn't focus on one MMO - too much time investment required. How about an MMO (massively-multiplayer online game) for life on the web?

I was living with M (name and likeness redacted by request), a creative writing major at USC. M and I teased out ideas for role-playing game mapped to web surfing. So you can earn points for doing what you're already doing!

Mentored by a number of agreeable engineers, I rigged up a single player prototype, using a Firefox extension that saved my web history to a file, then parsing that web traffic with PHP to give myself Dungeons & Dragons-style attributes. Surfing a lot of Flickr gave me high dexterity but lowered my constitution. Surfing the Wall Street Journal raised by intelligence but lowered my wisdom. I hand-coded game values for a few sites. M made art for each of the attributes based on the paintings of Alphonse Mucha. Livened up with some visuals this prototype was enough to present at a conference Aula summer 2006 in Helsinki Finland.

Game maven Alice Taylor was in the audience. She and Matt Locke at the BBC had a mandate to sponsor innovative educational interactive media projects. The BBC gave us a $20k grant to build a game to teach people web literacy, on the condition we hire a British programmer.

PMOG - the passively multiplayer online game

So now this became PMOG - the Passively Multiplayer Online Game. From Fall 2006 to spring 2007 we worked: myself Justin Hall as producer/web monkey, game designer/writer M, and a UK game engineer Duncan Gough whom we found through our friend Matt Webb.

The BBC grant and the confines of Scott Fisher's Interactive Media Division (IMD) at USC provided a fine place to incubate our idea. We were supported and stimulated and challenged by smart folks with a good deadline and feedback mechanisms driving us towards a playable experience. IMD Professor Tracy Fullerton helped us run PMOG paper prototype tests; between the IMD and Alice Taylor we had good oversight to keep us focused on building a coherent experience in a finite amount of time.

My last year there I was able to take a class "Business of Interactive Media" with a long-time game industry idol of mine - Jordan Weisman. He has started multiple successful game production companies based on new media and new markets. He shared data from his work with the class, giving me the spreadsheet tools to model the cost of running a business and launching projects. Now I could begin to imagine how to sustain and grow a new game idea.

We showed PMOG in the IMD May 2007 and we had 1,500 players! Here's a brief video where I speak to the reasons for PMOG before the IMD thesis show. Duncan built a basic Firefox sidebar to host a working web annotation game with character records, user generated play objects across web sites, much of the functionality that we'd initially envisioned for the game. It felt like a game, at least a fun web toy, with a chat room filling up with return players.

Here's some pictures and research documents from that early PMOG era.

As we shared our work online, people suggested we could raise some investment. We had a new type of game experience, a creative use of the Firefox platform. Maybe we were building some valuable part of the online entertainment landscape. Maybe we could be an internet startup! I had worked for some internet startups in the 1990s (HotWired, electric minds, Gamers.com), but I didn't really have a sense of how an idea turned into a business. Turns out you can raise money on an idea, and a prototype is even better!

This was mid-2007. The web was booming - new services were being funded and taking off. There was a sense that you could grow an idea into something large and it would make money from our large audience or it would be acquired by Yahoo/Microsoft/Google.

So we began presenting the game idea to venture capitalists in Silicon Valley. It was amazing to understand that this amateur game production could be seen as an economic vehicle.

First Round of Funding

Joi Ito was our first investor, an angel who agreed to come on if we could find other investors. Joi was a high velocity digital citizen and investor in a number of innovative fun projects. I had helped Joi set up his weblog when I lived in Japan in 2002, so we had some history of working online together. In 2007 Joi was an active World of Warcraft player: he inspired us to make a game that could involve players as much as WoW involved him. Having Joi signed on first set us up to find other folks.

Joi introduced us to Richard Wolpert, another angel investor with a history of work with Disney Online, Apple, Real

Networks - AAA quality companies. He pushed us to make the experience more accessible and appealing to a broader audience.

Bryce Roberts at O'Reilly AlphaTech Ventures was interested. OATV was a relatively new seed-stage fund, they were looking for companies that might be disruptive or challenging to existing markets. We believed we were opening the conventional massively-multiplayer online game experience to broader audiences, and we appreciated OATV's roots in open source and hacker culture. Plus Bryce had great internet literacy and a fun punk-rock attitude.

I remember sitting in his beautiful conference room with a view of the San Francisco Bay, and him asking: "how much do you think GameLayers is worth?" (theoretically,after investment). I had somehow forgotten to plan that answer, so I winged it: "$3 million." Bryce replied "how about $2 million" and I said without much hesitation "okay." So there we raised $500k on a post-money valuation of $2 million - $50k from each of the angels and $400k from OATV. OATV pledged millions more in reserve if we showed promise.

I graduated with my MFA in June 2007 and turned down a well-paying offer from Yahoo Games to architect a meta-game of badges and points across their service. The deal to fund GameLayers, Inc. closed in July 2007. M graduated with her undergraduate english degree in May and joined J.J. Abrams' production company Bad Robot as a receptionist training to use the 3D printer. M left that job after a few months to be Chief Creative Officer of GameLayers. Duncan our play prototyper became our CTO, I was the Producer and CEO - we were now three co-founders making a new online experience!

M and I were living in Culver City, near Los Angeles California. Mark Jacobsen and Bryce at OATV strongly encouraged us to move the business from Los Angeles to San Francisco, to be near more developer talent and to be near other internet companies for collaboration.

The money didn't actually arrive until September; one of the first great lessons: securing funds takes longer than any prediction or arrangement. That month we packed ourselves and our dog Pixley Wigglebottom into a UHaul and drove north to a large cheap loft in East Oakland, near San Francisco where so many other internet startups lived.

Now we were charged with building out our company. We spent $5,000 to buy pmog.com. We hired up to five people. During fund-raising I remember hearing that you could estimate the costs of running an internet startup at about $10k per full-time employee per month. That includes everything - overhead, salary, technology, and it actually works pretty well as a formula. So we went up to about 6 people, burning about $60k a month, which meant that we had a little less than eight months to prove PMOG could be a viable business and GameLayers could be a viable company.

In reality we spent the money slower than we predicted. It was harder to hire people than we expected. We saved money by working out of our home. We recruited folks we knew; our first employee web designer Cap Watkins was M's friend from USC; he slept in our East Oakland loft with us and our dog until he found his own place.

Duncan remembers: "when we secured funding from OATV and had a conference call with [an] extension-building company where we almost gave up a large chunk of the company in order to have them build our extension." It seemed like a fast way to get our game running, but we realized we wouldn't be able to iterate our own product if someone else built it. So we decided to hire up a team instead. As Duncan put it, that "...was a real challenge, and a test of our self confidence."

One of our first advisors was Ben Cerveny, a well-connected digital nomad who had worked between the web and gaming for years. Ben helped us find a playful contract artist/programmer Mark "heavysixer" Daggett in Kansas, we reworked our basic sidebar version of PMOG to have a more integrated HUD and a gourmet Steampunk look. We chose Steampunk because we felt like the genre was on the rise, and it was appealing to internet early adopters. We thought Steampunk would also help to distinguish us in the broader games marketplace - stylish playful nerdery.

Beta Launch and Roadshow

We launched PMOG in closed beta in February 2008 - people could play, but only with an invitation. This allowed us to gradually grow the game, to be sure our servers held up and we weren't getting overwhelmed with new customers. We had a signup form for people to request access; I would scan the email addresses for the words "venture" or "capital" or "investments" or "partners" or "fund" - this way I found a number of venture capitalists had signed up for the PMOG beta; I would reach out to each of them to see if we could book a pitch meeting.

February and March were a big roadshow - talking to the press:

- TechCrunch: "Play A Multiplayer Online Game While Surfing The Web: PMOG" by Michael Arrington
- MIT's Technology Review: "All the Internet's a Game" by Erica Naone
- Wired: "A New Type of Game Turns Web Surfing Into
- All-Out Information Warfare" by MJ Irwin

We presented "passively multiplayer online gaming" at conferences: Game Developers Conference in San Francisco (coverage), Emerging Technology Conference in San Diego, South by Southwest in Austin.

In our company IRC chat room, with people from the UK, Missouri, Kansas, San Diego and East Oakland, we worked to evolve our experience, about six of us. Two and a half coders, a front end person, a community support person, a producer, and a game designer/writer. Not enough engineers, but we couldn't hire many - during a technology boom engineers are scarce. I pitched in making web views as the Producer (the views and occasionally controllers part of Ruby on Rails' MVC).

Another major lesson from running a software startup: engineers are the dreammakers and everyone is in fierce competition for talented programmers.

M envisioned a virtual world on top of the internet: a battle between order and chaos, where we supplied the information weapons. For charitable sorts, there were crates for treasure to be left on web sites. For aggressive provocateurs, there were mines that exploded and shook your browser window when you hit their target URL. Surfing with our toolbar game on at times made the internet feel alive with play in the margins!

In the spring of 2008, it looked like we had a promising game service that was getting a lot of buzz. We were spending slower than we'd planned, but we were running out of money. So the game designer/writer M and myself the producer shifted to fundraising for about 14 weeks of March-June in 2008. This is a huge attention-suck: tracking down introductions and opportunities, developing presentations, taking feedback, corresponding, prioritizing, traveling, pitching.

I started off with a stupidly fat slide deck. I thought it was important to walk through people through the idea, the history of the idea, the market context, the team, the upside, the plans. But a rigid linear presentation was anathema for busy people who have seen a ton of pitches. Each VC had their set of questions and usually couldn't sit through more than 3 slides.

We learned to show fewer slides and then just talk with potential investors. Often these were smart folks with experience building companies so we had a lot to learn and at best it was a good conversation, when we learned to relax on the formal presentation. We kept a stack of slides in our appendix; if they asked about "competitor products" or "projected burn rate & operating expenses" or "DAU versus MAU over the last three months" we could pull those up.

There were some soul-crushing moments: emerging from a red-eye flight into a room with six hard-driving rich dudes who want to know why we don't have explosions or blood in our game. Or why we're not a platform for embedding other people's flash games. Or why we're so deluded about the size of the market.

We were intent on building one game, because our team wasn't big enough to support multiple products. We didn't want to spread ourselves across multiple products when we hadn't proven the idea and built a successful instance of this new kind of game. We thought we might build a platform someday - opening up our tech for anyone to upload their games. But we wanted to get the core experience bolted together first before we supported other developers.

Here's a PMOG Experience video we produced as we were going out of closed beta, circa May 2008.

Second Round of Funding

Early June 2008, one of the craziest weeks of my life:

- Saturday: Get Married near Yosemite
- Monday: Move from 1400 sq/ft loft to 400 sq/ft apartment

- Tuesday: Sign Second Round Term Sheet for $1.5 million
- Friday: We fire someone for the first time

In June 2008 M and I were married in Central California. Our investors and co-founder from the UK joined our friends and family betrothing the Chief Creative Officer and Chief Executive Officer of GameLayers. Wedding ceremony was Saturday. Monday we moved from our large East Oakland loft into a tiny apartment in an awesome part of San Francisco's Mission District. Tuesday we signed a term sheet with Rob Coneybeer from Shasta Ventures for a total $1.5 million round that we had just raised after a long roadshow together. Friday we had to fire someone for the first time. It was extremely intense running a startup, and being married to a co-founder / board member. Nonshop high-pressure togetherness. We put off our honeymoon and kept working.

Fundraising on an offbeat game on an unproven platform was challenging. Fortunately we found an investment partner who saw the potential in the widespread social experience we were building: Rob Coneybeer from Shasta Ventures. Rob joined the GameLayers board along with M, Bryce and myself - leaving an empty fifth seat we didn't worry about filling.

June 2008: Portfolio.com: "Having Fun, Making Money": "a startup turns the entire Web into a game, aiming to steer players toward and away from certain sites. Will advertisers play along?"

We raised another $1.51 million dollars - $1 million from Shasta, and another $500k from OATV. Our two angel investors Joi and Richard had the option to put in another $50k each, but they chose not to. GameLayers was now valued at just under $5 million!

GameLayers Round 2 Term Sheet with Rob / Shasta - a triumphant document for us.

We were running on fumes when the second round of money came through. Now, August 2008, the money was in our bank account, and we ramped up plans to expand our operations, reach into new markets, improve our featureset, and draw in a ton more users! Rob showed up with a lot of enthusiasm and ideas for the project, giving us a boost of energy. We looked forward to focusing on our game-making again, after months on the road!

Kicking More Ass, Sooner

Our first venture capitalists OATV primarily work with nascent startups. Bryce focused on helping us build a team, and helping us get a rhythm building and releasing software. Then Bryce helped us present ourselves as a valuable company for other larger VC investment. Our second VC Shasta was accustomed to blowing out a product with promise, making it grow seriously fast. Rob wanted to see progress and drive momentum.

One year after we started, we finally stopped working from home and moved into an office. 76 2nd Street, the top floor of an old building for about $3100 a month, space for 14 people to grow into.

One of the biggest challenges for me was learning to prune my stories. I would get a call from Rob at 9.30pm on a Tuesday, how's it going? And I would immediately start listing the bugs that were on the top of my mind, the problems we were working to solve. Looking back I believe Rob wanted to know how we were doing strategically, and maybe a sense of confidence from me that we were kicking butt overall. I've been a journalist for years; in this case, my urge to tell a compelling human drama was not necessarily compatible with "here's a clear path to acceleration and growth." VCs aren't looking for spin, but I think they want to know that if you're thrashing on bugs it's part of a coherent larger vision.

Becoming a CEO and raising this money was committing to kicking more ass, sooner - an exhiliarating proposition. And because I believed in our idea, I wanted to drive it forward as fast as possible. I charged in, learned everything I could from books, advisors, news, and worked to be the best possible CEO Decider Visionary Leader Capitalist.

But occasionally I was too in the weeds. Scavenging on Craig's List for deals on monitors and printers to save $100 when tens of thousands of dollars of employee time was hurtling forwards towards a moving target. Running a startup is a crazy range of affairs to manage: the unceasing search for engineers, customers having weird issues with common computer setups, investor relations, the mood and productivity of the folks who gave up other opportunities to join our cause, the amount of cash in our bank account, our landlord refusing to install heaters so employees are buying USB-powered heating mittens, state-by-state tax and liability law, we're not paying ourselves much and we're eating out constantly and we're paying rent to live close to the office in one of the world's most expensive cities so personal money is tight, someone else just raised 3x as much money as we did to do the same thing with a larger team. And OMG the entire world

economy is cratering!

CEO Search

In November 2008 the board met for dinner at a SOMA restaurant called Salt House: M, Rob and Bryce. Rob and Bryce suggested Gamelayers needed a new CEO, and my wife M agreed: we needed a firm hand to turn this into a business, someone who could raise confidence with another round from more heavy-duty investors. Someone I could learn running a company from. I had gone from game journalist, to game prototyper, to game company pitchman, to game company head, and now I had some reckoning to do. I was not providing leadership for GameLayers that left my board confident in the future.

Time for me to search out a person better suited to my role at GameLayers! That was an intense ego slam. We spent $33k (first installment on a $100k retainer with the Cole Group) and now a parade of CEO candidates came through. It was a fascinating experience - to meet guys who believed they could make our company evolve and succeed. After interviews, I didn't think many of them were smarter or better-suited than I was. I appreciated the chance to browse smart potential CEOs, but I also occasionally resented spending upwards of 20% of my time trying to replace myself. It seemed like a confidence game, could I learn enough and re-apply for my own job, owning the confidence of my investors and my co-founder? Or was I better off in another role, leaving the brass tacks of business to someone else so I could focus on building a successful product experience?

The global economy tanked before we could find a replacement CEO; I stayed on by necessity. We were running out of money to pay salary for a big new CEO, and with the stock market plummeting, we couldn't expect to raise another round quickly - we had to make PMOG succeed with what we had.

Primary Objectives

Fortunately, through our Executive Search we found an amazing advisor Michael Buhr. Michael was an executive at eBay and the CEO of StumbleUpon, a Firefox toolbar helping people experience the breadth of the web. Michael had worked at a 1990s game console company called 3DO but he wasn't deep into games. Instead Michael knew about motiviating people and executing on business objectives. Michael was a great manager, a great leader, and he became our advisor and executive coach. For a few hours a week by phone or in-person, Michael helped us shape the short story of the company, putting forward a vision for concrete action. He sat down with the founders and asked hard questions so we could substantiate our plans.

Michael would have been an ideal CEO for GameLayers - he understood the Firefox toolbar space, he would have given us freedom to run product within business constraints, he was fanastic to work with - we left every meeting with him feeling smarter and more able to have fun at a challenging job. But Michael already had a job and wasn't drawn in by the opportunity to lead GameLayers.

Michael explained to me that a CEO needs to be able to state his or her primary objectives at a moment's notice. What is your core focus? Use your core focus as a razor - everything else is distraction.

In late 2008 we wanted to get more users to play our game, get them to be more active once they started, and get them to make friends and invite more folks into the game. We called these three things Accessibility, Engagement, Social. Then Michael asked us to name 3-4 efforts we were undertaking to move each of those forward. Here's our December 2008 GameLayers strategic priorities document, the type of focused thinking we developed with Michael's help.

The Nethernet

We renamed our game from PMOG to The Nethernet (TNN): perhaps calling it a place instead of an acronym would make the product more tangible and accessible. The Academy of Interactive Arts and Sciences nominated PMOG for MMO of the year for 2009; we lost to Blizzard's World of Warcraft. We had more attention, and we scored a huge distribution deal - being a recommended Firefox plugin by Mozilla. Suddenly we had a ton of traffic, and our servers began to groan.

We had an active thriving community. One of our early hires was a part-time community manager: Joe Wagner, a San Diego State University student with a passion for online game community. He helped surface issues from players and communicate our intentions to the passionate fans. We were consistently honored and delighted by the folks who shared

our game with us. They made a podcast, the Tubenauts. They ran a gambling circuit using our currency. They made thousands of guided tours of the web. They banded together to create games within our game. They stepped up to serve as Stewards: guides, ambassadors, chaperones and unpaid expanders of our game world. They supported us, expanded our vision of what was possible, and made us believe that we were helping bring beauty and delight to citizens of the web.

These days I would do a full day of 10-12 hours at the office, then have some dinner, maybe a beer and a bong hit, and work another four hours or so. I avoided caffeine nearly throughout; light substance use in the evening allowed me to unwind briefly and then plunge back in with renewed vigor. Some times late at night I might sneak in some work on things like GameLayers promotional vids to post on YouTube and hopefully find new audience

I wanted to make an action trailer, giving an urgency and excitement to The Nethernet. Searching online music archives for a free soundtrack M and I found a song called "ИНТЕРНЕТ ПИЭДОБОЛЫ" by a Russian band called Distemper with a serious propulsive drum track opener.

The title was offensive perhaps if you translated it. The lyrics (translated from Russian) are about people getting stuck-up through overuse of online networked communications. What a coincidence! Internet commentary and a great drum track! We would use the instrumentals for our internet warfare game and we would attribute the song in the original Russian title. It would be an easter egg, if someone looked it up: they'd find out the PMOG video soundtrack was a song called "Cunts on the Internet"!

We told our colleagues and they all felt it was a terribly bad idea to associate this with the company and the brand. So we decided to play it safe. I published the trailer with no soundtrack and moved on.

Now I'll put it up here, mostly because we picked the music before we knew the title - when we translated it, our minds were blown. It was sad to rip the music out to get it released on time, with no awesome soundtrack, and no bawdy cultural commentary :-(

Let's New CTO

Working long-distance with our Dartford, UK-based co-founder and CTO Duncan Gough had allowed a round-the-clock approach to running GameLayers and building PMOG. We would chat in IRC continually, just before each of us went to bed, and check back in, in the morning. All of us chained to our laptops in different time zones, watching the servers hum, hacking on code, design, google docs, keynote slide shows, photoshop, textmate, ruby on rails, capistrano, memcache, highrise, github, hiring, staffing, management, strategy, focus.

After two years of that we needed to have GameLayers working in one shared location. Getting Duncan a visa to move to the US had been an expensive challenge, and Duncan couldn't move from the UK to SF, so we parted ways in March 2009.

Now we had a lot of traffic, overwhelmed servers, and no CTO. I devoted all my time to finding a replacement, and a recruiter lead us to John Cwikla. He had the exact experience we felt we needed: he'd taken products with some potential, and some stability issues, and gave them a solid software foundation from which to grow. A veteran of multiple startups, Cwikla hit the ground running as CTO in May 2009. We revised our infrastructure some, and prepared to exit "beta" - to launch a product to earn money.

Revenue?

PMOG / The Nethernet was expensive to run. We had built prototypes to prove ideas; we hadn't built something that was conservative with computing resources. We had thrown money at web hosting, figuring that our hosting costs would only grow more expensive if we were successful. Cwikla rigged up The Nethernet to run using the cloud, basically preparing the software to scale inexpensively. Unfortunately we were chained to a ~$20k-a-month dedicated cluster server and service contract with Engine Yard from before his time. I worked to lower our monthly bill but Cwikla's code couldn't immediately trim our overhead.

Mid-2009 we were pressed up against a wall, watching the money remaining steadily decline. Sad times - we took salary cuts, we laid people off, including our talented, optimistic UI/UX designer Kristin Neinhuis, and a promising writer

Seraphina Brennan. We had tooled around with innovation, piling features into our web browser game. But when it came down to it, we hadn't baked-in monetization. We had a few ideas for earning revenue from the Nethernet; based on our small traffic size and custom platform, selling ads seemed to be an uphill battle. Plus the most popular Firefox extensions were about avoiding advertisements (Adblock Plus). So we thought, we'll give people the basic toolset for free, but we'll ask them to pay for the more crafty and crazy tools. Microtransactions!

We called the currency "bacon" which was a sort of inside joke in our player community, and we had fun talking about bacon as a currency: "bring home the bacon", maybe units of currency could be called "bacon bits" and so on. We launched monetization in a hurry and made a few key mistakes: we took away powers players had already and started charging money for them. We didn't give players much warning.

Selling bacon, monetizing the Nethernet was an abject failure. Plenty of vocal players hated what we'd done. Some folks understood why and were generous - one woman even ordered us a mega-pack of beer in congratulations. But there weren't enough spenders. We made about a hundred dollars a day the first few days. Within three weeks of launching monetization, we were averaging under $20 a day. $20 a day is $600 a month; two orders of magnitude off from where we needed to be. Considering we had a burn rate over $60k a month, we were profoundly far away from having a sustainable or profitable game studio.

When we launched PMOG we called our new players "Shoats" a word we discovered that meant "young pig." Ultimately with The Nethernet we weren't able to turn the Shoats into Bacon!

One Life Remaining

We felt burned by the Firefox toolbar market. Reading stuff like "who would pay for anything in a firefox toolbar?" from our players meant that we felt like we were a long way away from sustaining our company on that platform. It had been a gamble to build the first big multiplayer social game in a browser extension, and it looked like fail: the world wasn't ready or didn't need what we were building.

So we had about $500k remaining at this time. I drew up plans to fire everyone except one programmer, to see if we could keep things running and figure out where to find some more investment or revenue.

Marc Adams was a former U.S. Navy SWCC who was accustomed to 30 hour coding marathons at least once per week. Even working remotely from St. Louis Marc was so plugged into our pace that it felt like we could maybe sustain the company on him alone. Marc was profoundly committed to his craft; he was an inspiring engineer to work with.

Michael Buhr suggested we look at the last $500k as money we could use for a another startup. Maybe we if we had a small slice of a much larger pie: we had a functioning kick butt web games team, and we refocused on the most profitable market within our reach: Facebook games.

This was June 2009 and social RPG like Mafia Wars were tearing up the charts and making good money (a game called Farm Town had just debuted at #4). We were a Ruby on Rails shop, we didn't have any Flash experience, so we decided that we could build a social RPG in Rails in a few weeks. Then we could skin it to target different demographics. We figured that we could have multiple games up on Facebook within a few weeks.

Facebook: Dictator Wars & Super Cute Zoo

So we shut down our toolbar game PMOG/The Nethernet because it was a huge drain on resources and focus. That was sad, but it was easy because we were losing money so fast. In the absence of PMOG, some of our most active and frustrated players went on to create Nova Initia, another browser toolbar MMO of playful annotation.

Meanwhile the five of us, Justin, M, Cwikla, Alex, Marc, we built a social RPG game engine that could be rapidly re-skinned. For the first skin we chose a geopolitical theme - taking mafia battles and doing them bigger and more tough: Dictator Wars. We made the theme cartoony to sanitize and parody the horrendous political content.

I spent ten weeks reading books about Dictators and writing out content for our Dictator Wars content spreadsheet - it was fun to be a bit of a writer again. From books like Under the Loving Care of the Fatherly Leader by Bradley Martin and The Emperor by Ryszard Kapuscinski I learned terribly sad things about the worst possible violations of trust between leaders and citizens. I had dreams about dictators for years afterwards. I amused myself by twittering secretly as the_kim_jong_il

scheming to build a viral social media campaign for our twisted game.

We didn't want to fail as a studio because we'd only targeted one macho audience. So we went immediately to work on a second game, targeting women: Super Cute Zoo. We thought we could add some Flash pets and an improved UI and attract multiple audiences to the basic spreadsheet RPG gameplay.

We five people launched two Facebook games in 14 weeks. It was a long march of long days, after years of having made one sort of game, we had to start all over to make other kinds of games. We succeeded at our mission, we rebooted our studio! We got some nice press coverage from Inside Social Games: The Road to Dictatorship. We were contending on the most exciting new platform for online games!

We had to learn about Facebook fast. For example, in 2009 each Facebook user got a certain number of invites per day. Because we were fiddling with our game, we had our allocation reduced from 30 to 4 invites a day. That was a huge penalty, a giant growth slowdown for weeks when we only had months to live.

We spent a little bit of money on pay-per-click install advertising. But we soon learned that other games were spending ten or one hundred times as much money. We were advised to spend $50k our launch week, to place on the charts and get noticed. We didn't have that kind of money, and unfortunately we didn't have that kind of time.

Facebook games were changing fast. Increasingly farming games were all the rage. Players were expecting higher levels of interactivity. More polish. Our zoo-themed game seemed strange as a social RPG - any game that promised you a chance to build a zoo wanted to be point and click, instead Super Cute Zoo featured reading on static web pages.

Here's a stats snapshot, mid-October: Dictator Wars has 28k registered users, with 1700 a day playing (about 6%). Average revenue per day was $53 - about 3¢ per daily active user. The game was growing by about 100 people per day - a paltry .3% growth rate (damn our constrained Facebook invitations!). $50 a day is about $1500 a month; we needed 40 times that or more to sustain our company. We had busted ass, and we'd put something up on Facebook, but we didn't have the time or money to help it evolve into a successful property. With online social games, launching is just the beginning of the hard work.

The Last Ditch

As part of the second round of funding, Shasta and OATV had secured the right to purchase Warrants to the tune of a few hundred thousand dollars. Shasta wanted to be able to buy a larger part of the company at the Series A price if we took off. Late 2008 we voted to extend their right to purchase the warrants, in case we needed the easy cash injection. During 2009 we aimed to be worthy of these cash injections but by late 2009 there was no additional money coming from our investors.

I spent much of the fall of 2009 calling people I knew who ran social gaming companies, seeing if I could find anyone to buy us, to buy our assets, to hire our team - to provide job security for our employees and some return for our investors.

I even worked up a patent application for "System for Sharing Tasks between Players of an Online Game" to see if that might increase the value of GameLayers and help us find a lifeline to keep making games.

We had some acquisition interest from larger social gaming companies. But week-by-week the social RPG format was losing momentum in the market against the mighty Farmville. Basically, I could see people thinking, "why buy this company when their games aren't growing that fast - in a few weeks we'll try to hire their engineers maybe."

October 2009 I was invited to a GameOn Finance conference in Toronto. I blame Jason della Rocca, a relentless connector of people and ideas. Participating there was cathartic - I really enjoyed a chance to try to save other entrepreneurs from my problems and get inspired by other business experiments. Some nice folks interviewed me and posted it on YouTube.

November 2009 we pulled down our two Facebook games - they weren't making enough money, and we couldn't commit to long-term service. We took the remaining money to pay off the company's obligations, and arranged a decent severance for our people. We laid everyone off, and took a little bit of remaining money to put The Nethernet, our original game back online to run quietly at thenethernet.com.

Aftermathematics

What a sobering time! Begging people to pay pennies on the dollar for years of work. Begging people to hire away your best people. And then failing, and calling people who had given so much to this company to tell them I had to stop paying them, and soon they'd have no paid health insurance.

Starting a company is stressful, running a company is stressful, winding down a failed company is stressful. We needed a break, but we were broke. So we cashed in airline miles and spent on our credit cards to finally take a honeymoon, 18 months after our wedding. December 2009, three weeks of inexpensive tropical luxury on an island in Thailand. January we returned to the San Francisco area to seek our fortune and pay down our debt.

By February 2010 M got a job as a Producer at Playfish, a social / Facebook game company. M hired one of the remaining GameLayers engineers to work at Playfish: Alex Friedman, a persistent gamer/programmer who we hired out of college based on his combination of game design and engineering chops. I helped our tireless engineer Marc Adams find work with Facebook-game maker Turpitude Design. In March 2010 I got a job as a Producer at ngmoco:), a mobile social games company, working with an old friend Alan Yu. ngmoco:) soon hired our community manager Joe Wagner.

In mid-March 2010 I presented a version of this story at the Game Developers Conference, announcing that we were going to open-source our multiplayer client and server software. Six months later the code emerged in the public domain. Maybe it can be of use to some budding online social play experiments!

- https://github.com/gamelayers/PMOG-OS - the server code for PMOG / The Nethernet
- https://github.com/gamelayers/PMOG-FF-EXT-OS - the client Firefox extension code for PMOG / The Nethernet

Terry Thurlow was our consulting Finance Director. She was recommended to us by OATV when we raised our first round - paying her an hourly rate would ensure that we had all our paperwork in order for future financial events. Having a consulting finance director was a great aid, from day one until the day the lights went dark in our bank account. Terry had experience at Industrial Light & Magic and CollabNet, an open source framework, plus Foxmarks - she was well versed in the many various trauma of business operations and it was always entertaining to run numbers with her. Plus I really didn't have to worry about the books throughout.

M and I decided to get divorced in October 2010. I would suggest to anyone that if you're in love with someone, the feeling is mutual, and you decided to go into serious business together, you should immediately commit to ongoing couples counseling with a quality therapist. You're going to need professional third party help to calibrate and prioritize your relationship through the stresses of business. Holy Smokes it's intense.

Looking Back

Founding a company and developing a business was a wild fast education. By comparison, grad school was a relaxing time to explore ideas and make art. Running a startup was constant adrenaline: you must learn while doing. I'm grateful to my business partners for their patience during my learning time; I'll share my takeaways here to maybe help fast-forward another entrepreneur.

We built it all, ourselves

In 1994 I worked at Hotwired, launching one of the world's first commercial content web sites. We had to build everything. One of my co-workers there was Brian Behlendorf who was helping to write the popular web server Apache in tandem with launching hotwired.com. We helped determine the initial size for ad banners on the web. We drove business to Organic, one of the very first web advertising companies. Everything from scratch, on expensive Silicon Graphics server hardware. Hah!

Sixteen years later you can borrow social infrastructure from Facebook, rent a scaling number of machines from Amazon Web Services. Use Google to host your corporate email, documents, spreadsheets, presentations for free. House your corporate memory banks on Highrise for a small monthly fee. There's a plugin or social service for nearly any human interaction you might want to run; you just need to provide a new sort of glue: mixing and matching web services to come up with compelling new ways for people to express themselves, to spend their time and their money.

We started building PMOG in 2006, launched GameLayers in 2007 - that was just two years before Facebook became the dominant social plumbing for casual online entertainment. We built everything for ourselves - user profiles, friend connections, message boards, invitation systems. That took up a lot of our time - if we had started in 2009, we would have used Facebook as plumbing for our social graph and focused on what made our game fun.

Also Cloud Computing emerged in 2008-2009. Perhaps the Cloud could have saved us from our huge hosting fees and worrying about how many servers to commit to. But we were committed to a non-Cloud architecture and that seriously chewed into our budget.

I feel like online service-building has gotten so much easier since 2007 when we started GameLayers; it's fun to imagine this tech being even more accessible in another few years! Hurrah!

I Learned

Being CEO taught me to focus on a few key priorities. I learned to be ready to present a focused summary of our biggest opportunities at a moment's notice: here's who we are, here's our big vision, and here's how we're building to succeed.

I learned that companies are started by ordinary humans with big ideas, who convince other humans who have money to support that dream for a finite amount of time. I learned that people with money also have ideas, and its important to understand accepting money means you're going to be spending time aligning yourself with them.

Being accountable to my board of directors on a monthly basis was helpful: I had to pull my head up to look at the business objectives and discuss them with a smart committed crew committed to this shared undertaking.

I learned that the more of your idea you can build without financial support, the more leverage you'll have when it comes to raising money and attracting interest.

I learned that $300 / day is a great "lifestyle business" - $300 / day supports one or two folks making something they believe in, catering to a particular market. But $300 / day is death for a 5-9 person startup. If you've raised venture capital, you're aiming for $3,000 / day and $30,000 and beyond. Sometimes it seemed like maybe PMOG would have been a better "lifestyle business", meaning if we had prototyped and then started making money off the bat, we would have pulled down maybe enough to support the three founders at a smaller scale with something manageable. We could have grown PMOG organically, gradually, at a breathable pace! But we decided to go for an amazing accelerated roller coaster ride instead, to pour gasoline on a small campfire.

I learned that you can prototype an incredible viable product with people in far-away places. But it's hard to move truly fast when you're not in the same time zone. Ultimately being in the same room with people is a profound accelerator for collaboration and productivity.

I learned that engineers are the most important part of software startups, and hiring a lot of engineers is a good way to ensure you can evolve fast and increase the value of your organization. I learned there's a metric shit-ton of competition for engineer hiring, it's laughable. Every startup that has a web site with a jobs section with "engineer" positions listed. Fierce competition!

I learned to value the chance to meet with CEOs from other software startups. I would drive to Palo Alto or Mountain View to talk to these people facing similar struggles. It felt like community, I got great wisdom. Usually I tracked them down because we shared an investor or a common friend. Several OATV companies were funded at the same time (Get Satisfaction and Instructables) and we ended up having group OATV-funded executive dinners every few months - great camaraderie with people as busy as we were.

Alan Yu was a friend of ours; he was working at Electronic Arts at the time. He listed to us describe our product PMOG over sushi one night in SOMA. Alan suggested that between gameplay, theme and platform, it's advisable to limit innovation to one of those three when making a new game. With PMOG, we were doing an unusual game (leaving information tools on web sites for other people) with an unusual theme (steampunk was not mainstream) and we were building a massively multiplayer online game for a platform that had never really seen much of a game on it before (Firefox only had a few very basic arcade games, and one attempt to build multiplayer pong). We were innovating on three for three: gameplay, theme AND platform. That's a whole lotta innovation in one product experience yow. Now I think more about focusing innovation, creating a more approachable experience for game players.

I didn't have much time for reading while I was in the midst of this software startup, but I read Jessica Livingston's book Founders at Work. Founders at Work was hugely comforting to me, as I read of all manner of small software business permutations; people who succeeded and failed and pivoted and dealt with all the complexities that were so a part of my life.

Daniel James was actually part of my education from an early stage - because he shared his data year after year with the audience at the Game Developers Conference, and then on the web through his site The Flogging Will Continue Until Morale Improves. Here's a sample presentation: "Metrics for a Brave New Whirled" - transparent about data most companies guard.

I've learned to see more of the entire ecology of media/software innovation: from the limited partners LPs who fund the Venture Capitalists VCs, funding and prodding the founders and executives. I've been a disgruntled employee who doesn't understand why a company doesn't hew closer to its ideals; now I see so much nested economic activity that I tend to bite my tongue before critiquing corporate actors.

I learned that distracting employees should be laid off immediately. Firing people is hard, but if you're a startup founder debating whether someone should be fired, they should be fired. There's such a limited amount of time with a focused team; in a company under 10 people one underperformer brings everyone down and ties up invaluable time. I learned to think of firing someone as "helping them find a more suitable opportunity" because if they aren't thriving at your company, no one wins.

Laying off people was a fucking serious painful thing because these people were contributing doing important work to drive the company forward, but we just didn't have the money to survive. It's a sobering, terrible thing to have to say "goodbye, fare well" to someone who you've asked to give up other opportunities to join your crazy mission that could be a great success for everyone involved if only we weren't months from cash-related asphyxiation.

First I developed the reflex to test "I have a cool idea" against "I wonder if it would pay for itself", and then "I wonder if that's just a lifestyle business or a real company."

What Happens to the stuff we built?

Through the company ending and the divorce, I Justin Hall now own the code, art, trademarks, and URLs for The Nethernet, PMOG, Super Cute Zoo, Dictator Wars and GameLayers. Wow! It's all worthless unless someone invests more time and money to make something compelling from those bits, bytes and ideas. Social games require paid servers to run, engineers to keep them alive, community managers to run events and designers to make new content. None of this stuff runs without constant tinkering. So all these assets are aging quietly on hard drives awaiting purpose; at least the code for PMOG is open source so it might have life beyond.

The Nethernet was up and running, paid through December 2012. You could play it with a copy of Firefox 3 and a link to http://thenethernet.com/. Bless 'em there was a handful of players up to the very end, three years since we worked on the game. For 2013 and beyond it will cost about $130 a month to host. Who wants to pay that? Paying down a year of credit card debt after my startup doesn't make me too inclined to support the Nethernet indefinitely.

Besides, there's no support for modern browsers, and no programmers ready to evolve this software. Releasing TNN as open source was a nice gesture but it didn't summon keyboard-weilding brainiac badasses ready to update this game.

Twenty years from now the "social games" of the 2010s will be screenshots but not anything anyone can play; without servers you can't boot up and run a social game.

I confess I didn't really play the Nethernet after I laid everyone off at the end of GameLayers. I actually switched browsers - I needed to clear my head and now the game only works in Firefox 3. Rebuilding it for Chrome / IE / Safari is a challenge; maybe we could somehow port it to a floating HTML5 pallette or some other web hackery. I considered a kickstarter-type crowdfunding campaign to keep evolving this thing. But then I'd have to start beating the drum for this failed kind of gameplay, at best buying it another few months of life before it needs to find funds to stay online again.

I'll be keeping an eye out for projects that make use of the open source code! And I'll be keeping out for people who evolve the idea of games embedded in our daily activites, and social play across information space.

Running a Startup or Having a Balanced Life

I threw myself into this startup. It felt like a combination of a cause, a lottery ticket, an identity, a vehicle for realizing a better me, and a better life for my family. Looking back I savaged a number of my relationships through inattention and stress. I might have made a better CEO if I had taken some time for hobbies, reading, chatting, travel - some life balance. But I can still remember life in the startup trenches; I felt so important, like my every minute could feed this hungry entity and make it more likely to succeed. I felt so profoundly responsible and I got off on that feeling.

After being CEO of a startup, being a Producer at ngmoco:) was so nice because I could just focus on making a product, to not worry about payroll and investors. Working just 60 or 70 hours a week feels like a relative vacation. Being CEO meant I woke up at 4am most days in sheer panic about any one of a dozen looming madnesses. Now I wake up at 4am mostly because I had spicy garlic noodles.

I feel like my churning mind lent itself naturally to running a business. I like to deliberate over potential future issues and opportunities. John Lilly was at that time CEO of Firefox, and we had some useful meetings. He advised me: "don't borrow problems" - don't worry about things you can't control. Lilly also suggested being a CEO is often about looking over multiple fires and deciding which fire needs water most because you can't ever put them all out.

I now understand that most businesses don't succeed. While success is rare, opportunities to learn and evolve are limitless. My investors were gracious to the end, explaining that they knew they were taking a risk when they invested, and they appreciated the work we'd done to make a good go of GameLayers. I see them every few months now.

Months after GameLayers ended, I heard about another failed startup - this one had been funded by family members. As a result of the startup failing, someone had a mortgage they couldn't afford, and there was bad blood between the businessperson and their supportive kin. So I'm glad I worked with professional investors; they shared valuable experience, and they could afford to take a chance on this idea without painful loss.

I felt supercharged being responsible for a shared destiny, and though it didn't work, I would definitely do it again with the right concept or team. Work is best when it's fun and meaningful. Next time I would want to have more balance in my life; exercise, socializing, family time. But maybe I'm just saying that because I'm older :-) I'm not sure whether I'd be a good CEO again; I'd definitely do better than last time. Mostly I'm excited to pitch in wherever I can with compelling people and amazing ideas on crazy challenging new networks.

Why did we fail? What could we have done?

We saw three competitors start up: RocketOn and WelloHorld and WebWars - each making some version of "browser plug-in based gaming across all the web". And our players were frustrated when we took The Nethernet offline and so they created their own PMOG-ish toolbar game: Nova Initia. None of these has cracked the code of widespread browserplay goodtimes as far as I've seen yet. Maybe browsing the web and playing a game are two/too distinct activities for most folks.

Flickr was often on my mind: the popular photo-sharing service from the 2000s that was purchased by Yahoo. Flickr started off as an attempt to make a complex web browser game; a sort of MMO for comparative literature majors called "Game Neverending." The Flickr founders figured out that photo sharing might be a better use of the tech they'd built. I thought about how we might adapt PMOG to other web sharing, but there was no ready product coming out of our toolbar. And we really wanted to make an entertainment experience.

Working at ngmoco:) has developed in me an intense discipline for "compulsion loops" - what are the core activities in a game that keep people engaged? How does one in-game activity lead readily to another? And what is the core fun at the heart of your game? Ultimately I believe PMOG lacked too much core game compulsion to drive enthusiastic mass adoption. The concept of "leave a trail of playful web annotations" was too abstruse for the bulk of folks to take up. Yes being a Firefox MMO was a weird and challenging hurdle to adoption, but people will go to insane lengths for a good time. If we had presented an amazing fun entertainment product with a clear path to fun, we would have had a larger audience. More players would have made it easier to raise money and make money.

We heard the feedback "I don't know what I'm supposed to do" from new players and people who tested it - we thought we could add features or somehow convince people to power through to become dedicated players. Looking back I

believe we needed to clear the decks, swallow our pride, and make something that was easier to have fun with, within the first few moments of interaction. That may have meant scrapping "passively multiplayer" as a game concept, or a toolbar as the mode of interaction. Those were pretty core to our initial business plan, so it took us a long time to see those concepts through to revenue failure.

Then we pivoted: we shifted strategy from "build the best game played across the web" to "build games to learn how to make money on Facebook." We brought our vision in from the horizon to our bank account. It prepared us employees for careers working in social games (4 of 5 of the last GameLayers employees were making social games for Facebook or mobile within 6 months of the company closing). The sector was hot and we learned it. But what if we had doubled down on our goal of building a big game? "Okay, so The Nethernet can't support itself. We're going to break apart that technology into something smaller, a simpler game to play across the web" or "We're going to take what we've learned about play across the web and make a big game you can play anywhere on your mobile device." Would we have found a sustainable path by pivoting towards another bold future instead of quick Facebook monetization?

Don't know, can't know - happy to live on, to see all kinds of new games made on new platforms by new dreamers.

Ultimately I'm honored and humbled by the chance I had to work with fantastic motivated brilliant people each day on dynamic complex problems, to take an idea and make it real, to test our theories on the human market, and to fail with as much support as we rose. We didn't win the capitalist gold medal but hopefully we generated some useful knowledge and entertainment!

Original article here: http://links.net/vita/gamelayers/#Why%20Fail

Startup Failure: Unifyo

Unifyo: Post Mortem - 3.5 years and $700,000 worth of lessons learned from a 'relationship management' startup

Unifyo (founded as Handy Elephant Ltd in 2010) was a startup in the relationship management space. Our vision was to enable people to build and maintain more relationships. We have built 6 different products (and gone through 6 pivots), all of which focused on automating daunting contact & relationship management tasks for users to create a outstanding user experiences.

It started with me building an Android app for business professionals which quickly became popular—back then, the 'personal CRM' space was hot. After a year of bootstrapping, Robin Message and Lewis Spearman joined full-time when we received investment from the Citrix Startup Accelerator. We subsequently built a 7 people team and increased our total investment amount to $700,000 in total, adding investors like Seedcamp, EC1 Capital, Firestartr, Angellab and Tom Blackie.

The final value proposition was a sales tool for new business teams at enterprise size B2B companies, which enables them to find introductions to leads and prospects from within their own company. We had deployed solutions in big professional services companies to even a Fortune 500 bank, but ultimately did not have the bandwidth to succeed with an enterprise sales approach. Apart from committing typical startup sins like not being focused enough and occasionally trying to scale prematurely, we learned a ton of lessons that we hope will benefit the next generation of startups—especially in the 'relationship management' space, as there is still a lot to be done.

1. Don't hesitate to go after big enterprises

Whenever you talk to people who have to use CRMs on a regular basis to maintain the contact and relationship data, you can feel frustration. The ultimate outcome in terms of analytics and insights should be good, the manual data entrey and clunky interfaces are not and cause low data quality. We aimed to build a great, highly automated user experience first, focusing on the end-users and SMEs with plans to grow into companies from the bottom up (like Skype, Yammer, Dropbox). We couldn't empathize with big corporations and heard only scary things about the long sales cycles. However, every company we have kept track of in the 'relationship management' space has either shut down or moved at least into the B2B space (as opposed to targeting consumers), e.g. Nimble and Contactually. Except Rapportive which has successfully been acquired by LinkedIn.

The problem is, as soon as a key feature requires data from multiple users and has to access key data (e.g. email or CRM systems), you will need the buy-in from the head of sales (who can get a whole team on board) who will need to get approval from the IT department, who have locked most applications down. Once targeted head of sales type people in big enterprises, there was plenty of interest and we got further in the process than we thought we would, but ultimately we validated this assumption too late in the process.

2. Have someone with Enterprise Sales experience in your (founding) team

Brad Feld wrote a great article on about "founder market fit", which made us realise (2 years down the line) that we might need to up our game here. As we focused on consumers originally, all of us were passionate about building great user experiences. None of us were passionate about sales, nor did we have sufficient experience.

When I started reading books and threw myself at it (with the help of some amazing sales advisors, including Analisa Sarasini, Ben Rainbow, Julie Cheneau, James Ski and Steven Wakeling), things started taking off and it felt like we had product market fit, but still, none of our commercial hires for marketing and sales had a big impact on our growth and revenues as we did not truly understand what qualities we should be looking for in them.

Plenty of companies do well with targeting SMEs first and having their sales team pick up inbound leads. As the right market for us was with big enterprises, we would've fared better with a co-founder who has at least a lot more commercial experience than the 3 mid-twenties of us.

3. Sales cycles are as long as they say. Make sure you raise (more than) enough funding.

After bootstrapping, we raised $700,000 over 2 rounds which sustained a 3 people team (with 7 people at peak time) over 2.5 years, in each round we gave ourselves 12-18 months of runway at a stretch (with a small team and limited bandwidth). Enterprises are as slow as their reputation, it will take 12-18 months to get your money if you know what you're doing. I'm not saying there won't be exceptions (especially if you have existing connections and a great reputation with your first clients), but you need to raise enough money to keep going while enterprises shift focus, people wait for approval and people getting replaced. You will also go through multiple rounds of iterations on your security and compliance mechanisms, all this will take a lot of developer time.

This won't just apply to your actual sales cycle. It took 8 ridiculous months to get our (finished) mobile app approved by Salesforce for their AppExchange. And even Salesforce, which is a CRM company and should lead by example, isn't always up to speed. After paying for the security review of our application, they forgot us for 2 months. Once we chased them up on it, they said 'it fell through the cracks'. Thats Salesforce for you.

4. Appreciate the number of edge cases when trying to automate something with organic data

Unifyo had a few genius, truly innovative parts to it that haven't been tried before, including identifying contact names on any website and injecting items next to them without breaking the layout (so people don't have to manually look up contacts but get made aware pro-actively if someone in their organisation has an existing connection to a prospect).

We did this for 2 reasons: Provide an outstanding user experience (inspired by Rapportive and other 'install-and-forget' tools) and to avoid being at the mercy of APIs due to our experience with Salesforce (and also because LinkedIn, Facebook and Twitter started restricting their APIs when we set out). Although our algorithms were solid, there was always yet another edge-case that extended our roadmap (which was tight already, as we never had more than 3 engineers). So even though we felt we had product market fit with an amazing user experience in demos, in practice our tool mostly underwhelmed users — and there was no easy way to test and plan for all the edge cases with organic data that was out there and always evolving.

An example? There are dozens of ways the name "Don Draper" would appear on websites (including "Draper, Don", Donald "Don" Draper, Don D.). Either avoid having to mess with this (although, for outstanding user experiences in the contact and relationship management space, sensing that all these refer to the same person will be critical as you scale) or dedicate (more than) enough resources to testing this and constantly analyse the cases your users face. Greetings from the Machine Learning space.

5. Define (read: write down) your company vision and values

Every founder has got a vision and values. Few are great at communicating and engraving them in their company culture. To me, that happened because the vision was bold and broadly applicable ('everyone manages relationships'), but also because I did not take the time early on to write the mission statement and my values down into a document every applicant would read before even applying. It would've saved us lots of time and hassle by setting expectations right from the start.

Thinking back, I'm not sure what my excuse was. Make sure you get it right — Buffer has created an amazing slide deck on their values which I recommend every company to check out.

So, what's next?

We have just shut the landing page down, although the web service will keep running for the time being (no guarantee if you're a user).

We still have the startup bug, and while Robin (our full-stack CTO with a PhD from Cambridge University) and Duncan (our amazing front-end developer) feel like it's time for a change, I'm still very much excited about the vision of enabling people to build and maintain more relationships. But mostly, I think there are a lot of areas in which new user experiences

which automate daunting tasks for users will out-innovate existing players. I am still looking to join a company which has reached the next stage, has a great product culture and wants to make their users' lives better. If you've got a Product Manager opening, I'm all ears: ben@unifyo.com

Duncan will broaden his horizon while travelling, Robin will join another young company (TBA). And Lewis, our amazing designer, is all ears about any remote and contracting opportunities. You can email everyone at the corresponding firstname@unifyo.com email addresses.

Last but not least: THANK YOU!!!

A huge 'Thank you' to all the friends, mentors, investors, former team members, applicants, and other people along the way who believed in our vision, gave up their time, made connections or put money where their mouth is. And also 'thank you' to the people who said that we're crazy and that our idea is useless, you kept us working hard.

The past 3.5 years have been an invaluable and (if we hadn't lived through it ourselves) incredible experience that has changed our lives. We are grateful to every single person we met and will be more than happy to give back where we can. As a wise person once said:

"Entrepreneurship is a debt"

Original article here: https://medium.com/@benfwirtz/unifyo-post-mortem-ba205ef05cc6

Startup Failure: Lookery

Couldery Shouldery

We're commencing an "orderly shutdown" of Lookery. We made the decision as late as possible without forgoing our responsibilities to the various stakeholders, including and especially orderly disengagement from our customers. We're honored by their attention and support and regret that we couldn't have served them even better and for much, much longer.

I left Wouldery out of the post title as it's not relevant. The only information of value is what other choices were readily available to us and which ones I look back on with uncertainty. This is probably not my last public postmortem on the lessons learned, but I owe everyone near-immediate disclosure on the things I can put my finger on now. I'm trying to keep a stiff upper lip during this process. Please avert your eyes if I falter.

As co-founder and CEO of Lookery, the buck stops with me and no one else. I hope to have the opportunity to work again with each and every one of the dozen people who worked at Lookery since it was founded in July 2007. I hope I did as well by them as they did by Lookery and by me. Please take this post as a recommendation of each and all of them and never hesitate to get in touch with me for details. The huge majority of them went well beyond the call of duty to try and make the company succeed.

So did Allen Morgan, the outside Director that represented the Preferred investors. There's no way we could have gotten this far or known how to behave well in the face of adversity without his guidance. He represented a great batch of angel investors on our Board and was one of the main reasons that they were so supportive of the company whenever I asked for anything. In May and June of this year, most of them agreed to participate in an inside round to keep us running through the next series of milestones. I cancelled the transaction a month ago, as I could not be as confident of reaching those milestones as 'good faith' requires.

Before delving into our market and product choices, I need to address our fundraising overall. On forming the company, David and I decided to pursue a low-cap model, postponing institutional investor involvement for the foreseeable future. It really sucks to be here right now, but I think it was the right call. Resigning politely at someone else's behest, as I've done twice before, did not improve any of the stakeholders prospects in either case. Unless Lookery reached an expansion stage, where the sales model was known and repeatable, I'm ever-more convinced that raising institutional capital would not have increased our chances of creating value for the existing shareholders. The company would have survived longer, but that's not the goal.

Going low-cap, David and I were careful to create a parsimonious and headcount-spare operation. Operating virtually was not a problem, nor was cutting corners on travel, schwag, and office rent. And, I'll still only work on cloud-hosted businesses. Those parts of Lookery were appropriate, repeatable, and did not bring us to this point. We did invest too much in building market share in the original ad-network business in the first half of 2008, but that error was a symptom rather than the disease.

Moving on to our specific coulda-shoulda product and market choices, there are three key moments at which a different and defensible decision might have made all the difference. In chronological order, the sins Lookery committed under my leadership were continuing our dependency on a large partner (March 2008), not knowing when to cut bait on a failing asset (September 2008), and building ahead of the market (December 2008). I and we made any number of other mistakes, but all the rest were correctable. Avoiding even one of the three big errors might have been enough to get us over the hump.

I believe David and I started Lookery in July 2007 in the right way and for the right reasons. Based partially on my F8-launch work for LendingClub (a company I'm thrilled to know) David and I decided to quickly offer a no-frills banner network for Facebook app publishers. We went from commitment to live service in well under two weeks using AWS and OpenAds, pulled in Rex Dixon almost immediately to manage the publishers and Todd Sawicki soon thereafter as we needed a real ad pro. Both David and I had been keen observers of, and vendors to, the online ad business from the outside, and Todd was the online advertising insider that completed the early team. By March 1, 2008, we were getting pretty close to a billion impressions a week, had moved the ad ops to Atlas, started spec'ing Lookery's targeting technology, and closed out a $1M convertible note financing that had been rolling in since October.

So far so good on using an ephemeral opportunity to create a company, but this is where I place Coulda-Shoulda #1. We exposed ourselves to a huge single point of failure called Facebook. I've ranted for years about how bad an idea it is for startups to be mobile-carrier dependent. In retrospect, there is no difference between Verizon Wireless and Facebook in this context. To succeed in that kind of environment requires any number of resources. One of them is clearly significant outside financing, which we'd explicitly chosen to do without. We could have and should have used the proceeds of the convertible note to get out from under Facebook's thumb rather to invest further in the Facebook Platform.

Coulda-Shoulda #2. Predictably and reasonably, Facebook acted in their own interest rather than ours. Their Summer 2008 redesign supported Facebook's goals elegantly but hurt our publishers and us in ways that became clear just weeks after we'd raised another ~$2M. At this point, we made a mistake endemic to startup people. We followed our natural inclination as problem solvers rather than getting out while the getting was good. If we'd sold the ad network the minute we understood that we could no longer make it successful, we would have saved a couple hundred thousand dollars in working capital. Plus, the ad network would have fetched three to five times its low-six figure sale price less than 60 days later. That's a million dollar mistake I made in a very short period of time. I should understand sunk costs better than this.

To give credit where it's due, Todd closed the sale of the Lookery ad network to AdKnowledge in less than two weeks from the moment we decided it had to go. I was off promoting Lookery's targeting system to European demographic data sources and social networks and was not even in the country during those two weeks.

Coulda-Shoulda #3. Once we sold the ad network, I fell into a bad old habit — persuading my team to build something before the market was ready for it. Oren usually saves me from myself in this regard, but I didn't pull him away from Mashery for the day or two necessary to diagnose the problem. Mashery is doing so well that I clearly could have. Lookery's Profile SaaS/universal cookie mechanism is far more economic and effective than cookie exchange systems in a world where ad media and targeting data are separate commodities. That world is a year or more in the future.

This is the fourth blog post that I can find from a Lookery exec in which the primary theme is early = wrong. I felt pretty good when I wrote this one in December 2007 when Lookery was still doing all the right aka solely tactical things; the issue was weighing on me last month as I started to wonder if Lookery's inside round was a bad idea; but it hurts the worst to read this post of Todd Sawicki's from the day we met four months before Lookery was even founded.

Within the bounds of confidentiality agreements and showing respect for the many people who treated Lookery and me well over the past two years, I'm happy to respond to any questions that arise. My replies may sometimes take a day or two.

Original article here: http://rafer.net/post/168541483/lookeryupdate

Startup Failure: Canvas Networks

Today my startup failed

No soft landing, no happy ending—we simply failed.

It's been a long four year journey, full of highs and lows. I am simultaneously incredibly proud, and incredibly disappointed.

I'm incredibly proud of an amazing team and all that they have accomplished. Our most recent product, DrawQuest, is by all accounts a success. In the past year it's been downloaded more than 1.4 million times, and is currently used by about 25,000 people a day, and 400,000 last month alone. Retention and engagement are great. And yet we still failed.

It may seem surprising that a seemingly successful product could fail, but it happens all the time. Although we arguably found product/market fit, we couldn't quite crack the business side of things.

Building any business is hard, but building a business with a single app offering and half of your runway is especially hard (we created DrawQuest after the failure of our first product, Canvas). I've come away with new found respect for those companies who excel at monetizing mobile applications. As we approached the end of our runway, it became clear to us that DrawQuest didn't represent a venture-backed opportunity, and even with more time that was unlikely to change.

I'm terribly saddened that this may spell the end for our wonderful community, but it's my goal to use what little money we'll have left after the wind-down to keep the service alive for another few months. However as of today the team has gone their separate ways, and our doors are effectively closed.

I'm disappointed that I couldn't produce a better outcome for those who supported me the most—my investors and employees. Few in business will know the pain of what it means to fail as a venture-backed CEO. Not only do you fail your employees, your customers, and yourself, but you also fail your investors—partners who helped you bring your idea to life.

In my case, I am extremely lucky and grateful to be partners with people who are simply the best. What separates the best investors is not how they help you when you're a rocketship, but when your ship is on fire and you're venting atmosphere. In this case, our investors have demonstrated what sets them apart from the rest—they've supported me throughout the ups and downs, and especially the downs.

With that said, life goes on, and the best path forward is not a wounded one, but a more learned and motivated one. I'm definitely not itching to start another company any time soon—it will take time to decompress and reflect on the events of the past four years—but I hope that if I do some day decide to pursue a new dream, I'll be in a much better position to. After all, I did just receive a highly selective, four-year education for a mere $3.6 million dollars! (I find humor helps as well.)

As for what's next for me, I honestly have no idea. This is the first time in four years I've been at a crossroads like this.

One thing I'll be doing more of is writing about my experience. Partially because it's therapeutic, but also because if there's a silver lining in all of this (and there is), it's that I can help educate others about a path fraught with hardship, but rewarding nonetheless.

I'm also particularly inspired by what Everpix has done by making so much of their story public, and hope many others will follow in their footsteps of radical transparency. I don't wish to glorify my failure, but it's certainly something I'd rather embrace than hide behind for the next five years.

On that note, if there are any specific topics you'd like me to write about, feel free to submit them via my Ask page. Or if you'd like to say hi in general, please don't hesitate to e-mail me at moot@4chan.org. Believe me—I'll have a lot of free time on my hands these next few weeks.

Last but not least, to my team—Dave, Eunsan, Alex, Nick, Shaun, and Jim—I am eternally grateful. To my investors—Union Square Ventures, Andreessen Horowitz, Lerer Ventures, SV Angel, Founder Collective, Chris Dixon, and Joshua

Schachter—I could not have done this without you.

And to everyone who has supported us over the past few years, from the bottom of my heart: thank you.

Original article here: http://chrishateswriting.com/post/74083032842/today-my-startup-failed

Startup Failure: Blurtt

Shutting Down Blurtt

Editor's note: Jeanette Cajide recently opened an office in Cambridge MA for Dialexa, a technology product development outfit that helps companies, governments and entrepreneurs design and develop custom hardware and software products. She spent many years on Wall Street and in technology consulting prior to launching Blurtt. Follow her on Twitter @jeanettec007.

Blurtt, which launched in its current form back in 2012, was a mobile app that allowed you to create their own digital expressions by allowing you to choose an image, add text and share either anonymously or to your network. Now in 2014, my beloved startup Blurtt is dead and all the assets are for sale.

As one of the cofounders of the company, I thought it would be helpful to walk you through the many incarnations of Blurtt and the challenges we had along the way in hope that other startups can avoid our mistakes.

Death By Too Many Business Models

Ideas are a dime a dozen; the difference is in the execution. Below is a list of business models Blurtt considered, built and attempted to raise money on:

Idea No. 1 in 2009

Web and mobile platform that curates photography and art from individuals and you can create and send your own postcard from your phone for $1.99.

Executed: Functional web platform that was live and running.

Problem: We saw a new opportunity (idea No. 2) and changed the business model but needed funding to iterate and scale.

Funding: Bootstrapped with investor interest but no commitments.

Assessment: While many postcard app startups have received funding, this is a lifestyle business. If you want to turn it into a fundable business, you need multiple revenue channels and products. The exits, if there are any, will be small.

Idea No. 2 in 2010

Web and mobile platform that allows you to send free postcards in the mail with an advertisement on the back.

Executed: Along with our web-based Blurtt-creation tool, we created a functional self-service advertisement portal similar to Facebook Ads that allowed you to choose your target demographics, location and gender. In order to send blurtts, we required Facebook Connect, which allowed us to get your data and match you with the advertiser.

Problem: At this point we turned off monetization for Blurtt and they could be sent for free. However, we had no ad partners, as advertisers wanted to see high volume, and our fixed costs (printing, mailing, hosting) started to add up without any funding or revenue. I had to shut down Blurtt.

Funding: Bootstrapped. Tried to raise money but feedback we received was that no one wanted to send printed postcards, and the name "Blurtt" was not good for snail mail as "blurt" implied impulsivity. The impulsivity of expression would be lost in the mail.

Assessment: A startup recently launched in this space, and I think they will run into the same problems. The idea is good but you are adding an impersonal advertisement to a postcard that is supposed to convey emotion. While advertisers are trying to learn how to read and convey emotion through advertisements, it will take a considerable amount of funding and users to make this business model work.

Idea No. 3 in 2011

Mobile platform of micro-gifting and greeting cards.

Executed: Spent most of the year raising money on this idea. We received an acceleration agreement from Archimedes Labs in Silicon Valley.

Problem: The money we'd need to raise to make this happen would need to be considerable. We decided to build an MVP instead which at least addressed one of the two ideas.

Funding: Raised our first $25K, which made it easier to raise more money.

Assessment: The MVP ended up being Idea No. 4.

Idea No. 4 in 2011-2012

Mobile platform that allows you to choose an image from the web (or your camera roll) and add text over the image and share digitally. The blurtts could be shared anonymously or via your user name. The idea here was to give people a platform to use images to better convey how they truly felt. (An image is worth a thousand words.)

Executed: Mobile app launched in 2012. We were featured in TechCrunch and Mashable's Top 6 Apps to Download for the Week alongside Angry Birds.

Problem: It is hard to force a Blurtt. When we were working through the MVP it felt like "blurtting" was something one just had to get good at. Blurtt became a content-creation tool of which only 0.02475 percent of the people were actually both funny or clever extroverts who wanted to share their blurtts with the universe.

Funding: We came close to closing a $1 million dollar seed round. When the investment discussion turned to who was going to run Blurtt and what team was going to build it, I had a falling out with the lead investor. I raised a $120K friends and family round and built and launched an MVP using those funds.

Assessment: For the niche who is original, witty and creative, Blurtt was the tool for them. Some of the content people started creating and sharing was vile and not at all the vision I had for Blurtt. We could not find product/market fit but had several ideas of where Blurtt could go. I did not have enough funding to iterate real-time. Blurtt started to fizzle away but I was going to give it one last shot.

Saving Blurtt…Again

I have a Joan of Arc complex so the only time I will give up is when they burn me at the stake.

We launched Blurtt (idea No. 4) in March 2012, and by September 2012 it was obvious that we were not getting the traction we needed to raise money. I convinced one of my business classmates to let me live with her in New York City as I tried to find a technical co-founder (i.e. free developer) who could help me with Blurtt.

At this point, I had my bags packed and I went to say goodbye to the team I had hired to help me build my MVP, a product development company in Dallas called Dialexa. Co-founders Mark Haidar and Scott Harper said, "You can't just leave Dallas. You are way too smart and too talented to be sleeping on floors and couches and begging people to help you. We will take over Blurtt and you become the product manager and you can work for us."

After making zero salary for two years, managing $120,000 to build a product that might have taken any other person 2-4x that amount to build, I accepted Dialexa's offer.

The Personal Side No One Likes To Talk About

I joined Dialexa and we were all happy that Blurtt was getting another chance. But I started to feel burned out. I was Blurtt's fearless leader, but the problem with burnout is that you become hopeless and you lose every aspect of your creativity. I would stay awake at night thinking about Blurtt (what to do with it, punishing myself for my mistakes, thinking about the people who hurt me along the way and said I did not meet the "pattern of recognition for success" in the Valley).

I'd go to work feeling tired and exhausted. I was burning the candle at both ends.

If I had not had two to three years of battle wounds to recover from maybe I could have bounced back quicker, but I was spent: emotionally, physically, financially and mentally. I asked my investors (who were thankfully my friends and family and understood) to give me a year to see if I could find it within me to save the company.

After reviewing more business opportunities within Blurtt I realized it was time to let it go. In the end, time is actually your most valuable resource.

Some Other Takeaways

Taking venture money is not the end all, be all. I remember getting asked questions like "Is this a lifestyle business?" like it was a bad thing. The reality is VCs look for big exits so they don't want to invest in your multi-million-dollar venture even if you're making millions. I kept on changing the vision to appeal to investors so I could chase the bigger dollars so I could build the startup I really hoped to build. In the end, the passion and magic was lost.

Do not launch a startup if you do not have enough funding for multiple iterations. The chances of getting it right the first time are about the equivalent of winning the lotto.

Remember why you started this in the first place and never lose sight of it, because once it becomes something you are not happy doing — you shouldn't be doing it. I wanted to change the world by giving people a way to express their true emotions.

I want to take this time to thank Blurtt's cofounders Laura Jensen Gurasich, Kuba Tymula, and Nikhil Sethi who never gave up on me. My team at Dialexa who stepped up to the plate when things got especially tough and helped me get an MVP out that dazzled the world, even if just for a short while. And to my investors and my family, thank you for believing in me. When you invested in Blurtt, you invested in me. This won't be my last rodeo.

Original article here: http://techcrunch.com/2014/02/16/shutting-down-blurtt/

Startup Failure: Manilla

Manilla, The Hearst-Backed Service For Managing Bills, Is Shutting Down On July 1

Account management startup Manilla says that it will be shutting down on July 1.

We received several tips about the shutdown today, then when I visited the site and tried to sign up, I was directed to this announcement, which says that the service will continue to operate as normal until June 30. After that, users will no longer be able to upload new documents, and it will not retrieve any new data, though documents will still be stored on and downloadable from Manilla until September 30.

"This was a hard decision given that, over the past three years, Manilla has won many awards and has been well supported by its valued user base but was unable to achieve the scale necessary to make the economics of the business viable," the company wrote.

Incubated and backed by publishing company Hearst, Manilla launched at the DEMO conference in February 2011 with the promise of giving users a single website where they can pay their bills and manage their accounts (and access them on mobile apps as well). The service opened to the public a few months later.

Manilla also partnered with AOL, which owns TechCrunch, to create an AOL Bill Manager last year.

The company said that all the account credentials that users had entered into Manilla (so that it could connect to other services) will be deleted on July 1, as will all other data aside from un-downloaded documents.

Original article here: http://techcrunch.com/2014/05/09/manilla-shuts-down/

Startup Failure: Pumodo

A startup postmortem with a happy ending? ..in Thailand

When facing the reality of a failed product and 3 months runway left of investors money, sometimes the best solution is to go to Thailand.

In early 2012 Thomas and I envisioned a future where recreational football players would use an app to share their matches with friends and family. All their history of achievements would be recorded that would become their online football identity.

Using mockup design we tested the idea at KB (football club in Copenhagen) with players and coaches. The response was very positive. So we gave ourselves three months to build the first version of the app and test with a group of teams.

At the time I could work full time as I just sold my previous startup. Thomas on the other hand was contracted at Scandinavia's biggest deal site to upgrade their whole backend. A tall order for any hacker. So building this app on the side would not going to be easy.

As I started working in the morning, Thomas would join me after his work and code till late after midnight. Those three months I think he slept 4-5 hours each night.

Clear vision and validation ironed out

Pumodo was launched in App Store in mid April with 5 test teams. The feedback was promising and the data showed that each user would in average add 6.4 players to his team. A very good growth indicator.

Interviews showed that spectators really loved the idea to get live score notification from the matches they were unable to attend. The math was that each player had in average 3 fans so a team had 45 in total. Often only 5 people attended a match so there would be 40 friends and family at home wanting to be updated on that match. The key would be getting players to track their matches.

A Skype meeting with senior level guys at Adidas confirmed that we were on right track in regards to our monetization model. They were committed to launch a campaign in our app once we reached a specific amount of users.

Thomas and I had the proof we needed. He just finished the contracting assignment so it was perfect timed as he could go full time on the project.

We also decided to work on the app the next three months in Silicon Valley. We'd had never been there before and it sounded like fun.

There we also could test the US market that was among the top 3 in size in regards to active recreational football players.

Do or do not. There is no try

In August we left for Palo Alto where we were invited to stay for two weeks for free in a cabin build in a backyard. The host was an awesome guy we met a year before in Copenhagen and one of the most esteemed growth hackers in The Valley.

From the comfort of the backyard cabin we moved to big city San Francisco and faced the totally hopeless situation of finding a place to live. We survived the first two weeks by renting office space at Tradeshift and sleeping on their couches at night. Not sure if they ever found that out.

Finally we managed to get a 9 m2/97 sqft room through our Y-Combinator network. Rent $1,250. Dang! But since we had free office space from our friends at OneLogin, our burnrate wasn't yet gone through the roof. But close..

We did the show and tell dance at Techcrunch Disrupt and DEMO. Networked at GigaOm RoadMap Con and YC Startup School. Got tons of advice. Met people with amazing experience. We felt we were in the nexus of entrepreneurship and innovation. And we probably were. But we also got tangled in the hype machine. Our business plan was changing every

week. We went from focusing only on football to becoming app for all sports. Think BIGGER was the mantra.

It was like our feet never touched the ground.

Our market research showed USA would not become our main market. Too fragmented and unfavorable league structures. Europe would be our target.

So in November we headed back to Copenhagen just in time for the Danish App Awards where we were nominated for Best Sports App and Best Functionality. With our feet on solid ground yet again we won as Best Sports App. Sweet!

And since we had been contacted by Danish business angels interested in investing, we felt everything was going as planned.

Winter is coming

One of the biggest disadvantages with recreational football in Europe was the long winter breaks. In Denmark it was 5 months. I felt like Ned Stark facing many years of winter. It rendered our app lifeless and desolate. No live user data to use to improve the app as majority of our users were based in the cold north of Europe.

Trying to remain positive we saw it as an opportunity to submerge into our hack cave to build the next big feature.

We wanted to solve the lack of ability to compare players across different positions. Current comparable metrics were goals and assist. It was not possible for a defender and forward to compare fairly.

The solution was an universal score based on players abilities to compare and compete against anybody regardless of age, level or position in lineup. Kinda like Klout score just for football players. The bigger impact you have in a matches, the higher is your score.

Clubs were offered a widget that showed Team of the Week based on the players' score and a leaderboard among the clubs players. We wanted the clubs to celebrate individual players weekly achievements with place on the clubs Team of the Week

The hypothesis was that if we can get a club to place it on their club page and promote it, the players would want to be on the weekly Team Of Week so they download the app and start tracking their matches.

Getting as close to our users as possible

In April we closed a $180k angel round and hired our first employee, Anders as a frontend dev. He was a great fit. Skillful, curious and a football fan.

From the desire to be in close day-to-day dialogue with our users, we decided to move our office to a football club (B93) in Copenhagen that was 1,800 members strong. Our office had only smelly locker rooms as neighbors. Perfect!

We were basically living with and meeting our users every day.

The coding of these two features took longer than anticipated which meant that we would not be able to launch before the summer holiday that is from May to August. The breaks between season started to really become a huge problem. How had we not seen this coming?

We needed users to test the new features. So that summer we went on a 7-day guerilla marketing campaign at one of the worlds biggest youth football tournaments, Dana Cup. We made a special Team of the Day webpage for this campaign. Promotion was with flyers and posters.

Despite personally handing out 10k+ flyers only 3 teams tracked their matches. The morale was still high but concerns started to arise. No panic. Yet…

If a platform fails at encouraging creation, it breaks down

When the new season started we had convinced a handful of teams at B93 to start tracking their matches. It started well

as the matches were tracked and the "Team of the Week" on the clubs webpage was generated.

But the conversion of teammates joining after invite was very low. After 2-3 matches the tracking stopped. Even those players who had been explicitly instructed by the club director to track, stopped tracking. We did multiple interviews with the players and the general response was "the idea is cool, but meh.."

We had created an Indifferent product. Oh, the shame.

Pumodo was a product where people wanted to consume the live matches, player data and history but players didn't want to supply it. The reward wasn't high enough compared to the effort.

But our biggest self realization was that we were not users of our own product. We didn't obsess over it and we didn't love it. We loved the idea of it. That hurt.

Coinciding with our time of woe, the Danish National Football Union announced they would launch a competitive live score app. With all their members already in the app it would get even harder for us to sign up users on our home market.

We had good ideas how to fix the situation but with 3 months runway remaining of our angel round and, once again, a 5 month winter break upon us, we needed radical ideas.

Further funding was not possible without a critical amount of active users and an proven product/market fit.

One does not simply pivot

Our conclusion was that if we would use remaining resources to try to fix Pumodo, the odds would be against us. Even with extended runway the winter break would make it very difficult to test and validate the new ideas.

So why not create a new football app build upon our current experience and code…

During the last year working on Pumodo we had noticed accelerated technology leaps in live tracking of pro football players during matches.

The ability to track and record 1,000+ live events per match created incredible rich data that nobody was utilizing properly. Live score services still served fans with simple and stale stat notifications in apps and on web.

So we looked at the market for live score football apps. No real market leader. But the market was growing fast as this seemed to be the last category that not yet migrated to mobile. We looked at some of the major apps and thought we can do much better.

But we needed more runway!

Extending the runway by bringing down expenses was the only solution. Denmark is not a cheap country to live and work in. Almost as expensive as San Francisco.

So moving to a much cheaper country made sense. We narrowed it down to Philippines, Indonesia and Thailand. After much consideration we chose Thailand as Thomas had a good friend who lived there and owned a small co-working space from where we could work and live.

As the dust had settled, we realized that this time we did not have to convince teams to input data. We could just buy it. Awesome!

From the ashes a phoenix arises

Four weeks later in mid November Thomas and I were on a plane to Thailand. Anders would join us later in January. We decided to bump Anders up to co-founder. He shared our passion for the new product, was prepared to take a drastic pay cut and we really liked him. Best decision ever!

The next months went fast. Our corner of the co-working space was quickly transformed into our new hack cave . We were spitting out code and design in a daze of strong coffee and thai Red Bull. The Pace was frantic and probably

prolonged the jetlag with a week.

We didn't care. This time we were building an app with ourselves as users. That just felt right.

We designed the ultimate football app and then scraped everything "nice" off. Left was a minimum viable product that could be build fast and get out in the hands of users asap. All hypothesis could then be tested and iterated on as we got feedback.

One thing we thought was missing among the other football apps was the ability to get notification for individual players. Get updates of just Zlatans achievements without following PSG.

This would become our unique value proposition.

Out of money but an awesome app richer

Last wages were paid end of March. As expected. With barely $1,000 remaining, it was time to get this show on the road.

On April 9th Apple approved version 1.0 of Champion.

We limited the launch to Thai App Store only. It gave us a contained environment to learn and iterate. To get users in the app we used most of remaining cash to buy cheap app installs through Facebook ads. We had 5 weeks left to test the app in before the football season ended.

30 days later we became no. 1 sports app in the Thai App Store.

Then Apple promoted us. Boom!

The launch had gone so much better than hoped for.

Out of money. Don't care. We're driven by the excitement of building, what we think, can become the best football experience on mobile. An app that empowers fans with great insight in players stats and history.

Soon it will deliver a live-score experience so powerful that we expect to see José Mourinho sneak peek at our app before making substitution decisions.

But right now we're laser focused on the biggest football event of them all: The World Cup in Brazil. To celebrate this we will add some Easter eggs that maybe will ruffle some Uruguayan and Brazilian fans.

Hope this story can become inspirational for entrepreneurs when struggling with a failed product and end of runway.

More posts will follow about our story and discoveries.

PS. If you want to help us, check out Champion and, if you like it, share it with friends. Thanks!

Original article here: http://fridriksson.tumblr.com/post/86584610871/a-startup-postmortem-with-a-happy-ending-in-thailand

Startup Failure: Admazely

STARTUP FAILURE: how it feels

Blogging to a following of what is probably no more a few friends, colleagues and business acquaintances is in essence either self-promotion or - in this case - a public therapy session. I'm hoping, however, that an entrepreneur or two might find this post and see that the grief and suffering they're feeling does not make them the only sobbing loser in the world.

My startup, Admazely, has gone bankrupt. It has failed. It's over. Done. We ran out of funding and didn't manage to raise more money.

I'm not writing this to get pity from anyone. Or to get even with anyone. Or to out anyone. Except for myself. Let me be very very clear: the failure that I'm writing about here is mine and mine alone. Were other people involved? Absolutely. Did their actions influence the outcome of Admazely? Sure. Was it their responsibility? Hell no! In the epic words of former Secretary of Defence, Søren Gade, the buck stops here.

I'll try and find the energy to do a more analytical post mortem with root-cause analysis and everything at a later stage. But right now, I want to share how my failure felt. And feels.

Admazely began informally in early 2011. Officailly, I began working on it in May (on a spare-time and hobby basis it was a few months earlier but my definition of when you start working on a startup is when you start doing it full time).

My eventual cofounders, David and Sylwia, joined in August. Fourth and final cofounder, Søren, made his commitment in March 2012.

We raised a seed round in April 2012.

And we formally filed for bankruptcy in May 2013.

This is the story of how that ride felt. What it feels like to fail.

First signs

Having a startup is an emotional roller coaster from the outset. Constant worry about finding cofounders, potential competitors, raising money, designing a product, building it, marketing it, finding a sustainable business model.

To me, the basic feelings associated with it are very similar to what I've experienced in previous jobs. I'm an "all in" kind of person, so I've felt a lot of responsibility towards my employers and my career in the past. But let me assure you that it pales in comparison to having my own startup. It's the same basic emotions but they get amplified by a factor of thousand.

I have interviewed hundreds of candidates for positions before founding Admazely. And sure, it's been really frustrating when I couldn't find the right person for the job. Or I'd be excited when I finally did. But interviewing people that I desperately wanted to be a good fit for Admazely, only to find out they weren't good enough was misery. Finally finding someone, asking them to join and be rejected gave reason to fundamental self-doubt. Am I not the mythological founder that rock star people are eager to join and work for free for a fraction of the company?

But it's also insane joy. When those amazing candidates agreed to join despite more lucrative offers from companies I admire and respect, I was exhilarated to the point of shouting and performing those celebratory scenes you only do when you're pretty sure no one is watching.

Where am I going without this? Well, in the general direction that you get a little bit numb to pain. You teach yourself to shake it off as quickly as possible and move on. When a lot of shit happened in a short period of time, I'd go into a state of mild stress and depression. But I'd have to force myself out of it. That led to this slight numbness which in turn led to a certain degree of insensitivity to signals from the outside. Because I was forcing myself to ignore them in order to not go under in pain, stress and anxiety.

And that slight numbness meant that I probably wasn't seeing any first signs of failure.

I obviously knew that we were fighting a multi-front war. We had a cash-out-date approaching from day one (end of April 2013). We had product problems. We had sales and marketing problems. We had people problems. We had process problems. Startups have problems. It's a grind - that's the nature of it.

So retrospectively I don't think that I was seeing any obstacles that I couldn't overcome.

Emotinal distraction

In February 2012 my wife, Kia, and I learned that she was expecting our first child. She was due in early October. Preparing for a baby takes up some emotional bandwidth. As it should. I'm pretty sure Kia thinks that it didn't take up as much as it should have. Even writing the section headline "emotional distraction" makes me feel a little guilty. Was my son a distraction to my startup? I told myself I could just add bandwidth but that turned out to be false. So preparing for a baby diverted my focus a little. But actually it was just a little.

Kia gave birth on September 27 and that's when the actual diversion of attention happened. For the last three months of 2012 I was doing a shitty job of being founder and CEO of Admazely. And an even shittier job of being a father and a husband. The latter is not the point of this post so I'll leave it out. It's probably the one I'll feel more guilty about in the long run. The former, however, made me pretty aware that I started being one step behind on many things. I felt it and my colleagues felt it. It felt like failing at the micro level. It was a tough time.

During the Christmas holidays I decided to get one step ahead again and I kicked off 2013 with an all-hands meeting setting a stronger direction and honing up to my mistakes in the previous three months. The team responded really well. They all knew it and could feel that we'd been getting lost due to my lack of leadership. And the next 45 days were some of the most amazing in the two years that I was doing Admazely.

The big blow

As mentioned, our cash-out-date was approaching. We'd joined the Accelerace program based on a recommendation from our chairman and representative of our lead investor, SEED Capital. The program does a lot of good things. The reason we joined was that it has a loan option for program alumni companies. The people in the program has to recommend you, you jump through a few hoops, you pitch and negotiate and finally you present to their equivalent of the infamous partner meeting - the Investment Committee.

We had done all of it and everything was teed up for the Investment Committee presentation. The program manager, Jesper, endorsed us vigorously, the program consultant, Christian, who had worked with us was an enthusiastic ambassador. We had spoken bilaterally to most of the people on the committee and had received confirmation that they would support our loan application. SEED Capital had agreed to syndicate the loan, further endorsing it. Our chairman was on the Investment Committee (yeah that definitely could be seen as a conflict of interest but that wasn't my headache) and had spoken to most of the members.

I had all of my ducks in a row so to speak. Or so I thought.

I went to present on Tuesday February 6 with confidence and less than two months of cash in the bank. It was snowing yet thawing. A disgusting day, typical Danish winter.

My presentation was ok. The mandatory Q&A afterwards was horrible. The only two people in the room that we hadn't gotten prior support from were skeptical to say the least. As I left the room I was shattered. And as my contact at Accelerace didn't call me later on that day I knew where it was going. My chairman didn't either. Not a good sign. I left messages and they didn't return my calls.

In the afternoon I had a speaking engagement in the other end of the country, pitching to a huge room full of potential customers. It was an absurd experience. I was crumbling on the inside but had to pose confidently. I had invited to of our customers to also speak at the event and after dropping them off I sat in the rental car staring at air. Crying. Feelings of fear, anger, self-righteousness and uncertainty overwhelmed me.

A million thoughts were racing through my mind. Guilt towards Kia for asking her to let me jeopardize our personal finances and then failing to pay back on her trust. Self-doubt. But it all just got worse from there on.

When Christian called Wednesday afternoon I knew the outline of the conversation before picking up the phone. I was oddly emotionally detached as we talked. Still kind of shell-shocked.

The worst part at this stage was telling my team the news. I had been 100% transparent about our cash situation and our process with Accelerace so they were expecting good news. I waited until Monday at the weekly all-hands. I should have spoken to them faster. Most of them knew me well enough to know what was up based on my mood and body language in the days leading up.

I told them that I would hustle to find other investors but that realistically it took more than the month-and-a-half of cash that we had left. Two and a half if we were all prepared to work up until a payday that we all knew would lead to bankruptcy, not pay. We all were, of course.

Piling on

December through March we were killing it in our sales team. After a lot of iteration we had finally gotten a small team in place that were converting leads into customers. The two core people on that team were Harriet and Nate, both Kiwis. A core element in my Hail Mary fundraising strategy was that we were finally getting sales traction and that we had found a model to build an international customer base through tele sales. Most Danish startups begin selling locally and then expand from there. We had chosen a pretty bold strategy (against the advice from our board) and gone for 'instantly global', meaning the bulk of our effort was in calling the UK. And for that you need native English speaking sales people. Hence my praise of Harriet and Nate (and Sophia, our Aussie sales supporter). We spend three months achieving what a successful startup like Trustpilot reportedly spent almost a year doing: making steady sales progress in the UK.

Mid-February - roughly two weeks after having our funding plan dissolve before my eyes - Harriet and Nate received news that they would not be granted permanent work visa in Denmark and were to leave the country within 30 days.

It felt like the most unfair thing that could possibly happen at that point in time. I had less than two months to raise about 3 mdkk with a pitch that now had zero chance of delivering if an investor should decide to invest. I remember closing my eyes when I got the news, laughing manically for like 10 seconds while just feeling dizzy. Harriet, Nate, Sophia and our "growth hacker", Daniel, had been busting their asses off for the last few months trying to crack the code to making it all work. And right there and then someone in SKAT decided that what Harriet and Nate were doing did "not qualify as the kind of work that a Danish citizen could not perform equally well".

What. The Fuck?!? I just couldn't believe it. The eloquence, snappy responses, mastering of nuances within the language that makes a really talented sales rep is based on intimate knowledge of said language. Their skill set were not some that a Danish sales guy could pick up from learning English in school or even living abroad for a short while. I was flabbergasted. And I wanted to scream. But even at that point I felt the need to keep a straight face to the rest of the team. I'm pretty sure I didn't manage. If only half the despair I was feeling shone through, I'm pretty sure my colleagues felt like packing up their shit and leave right there and then.

They didn't. Maybe they should have.

As I thought things couldn't get any worse, our CTO and cofounder, David, pulled me aside one day and told me that he and his fiance were moving back to Stockholm. It had been a long time coming. His fiance couldn't get settled in Malmö and wanted to go back. David had been postponing the decision but had finally chosen not to jeopardize his relationship. A good decision and one in which I support him 100% as a friend. As a cofounder - I found his timing to be the worst possible.

Anyone who have seen a cofounder leave their startup knows the feelings flying through my head at that point. Betrayal, sorrow, shock. Loneliness. Until that very moment it had felt as if David was the one person that would stay with me until the very end. But that wasn't the case now. The weight on my shoulders already felt heavy but it just got a lot heavier. And I knew that even if we did manage to raise more money, we couldn't realistically continue our upwards trajectory. Development speed would decrease dramatically and the driving force in getting shit done would be gone. No sales team and no CTO.

I still struggled to keep a straight face to the rest of the team, to my board, to customers and suppliers and to potential investors. And it seems I half-managed. But only half-managed. I didn't crumble completely on the surface but

underneath it was the most taxing time of my life.

I probably went and pitched somewhere between 10 and 20 potential angel investors after this point. And my pitch got weaker and weaker. Because I had lost faith myself.

I remember one meeting in particular that our chairman, Niels Vejrup from SEED Capital had helped tee up. It was with Jesper Buch and Ditlev Bredahl who had been dancing in the shadows for a while and were now ready to talk. It was a Skype call and I had tried to prep for the meeting. Tried to find ways to frame our situation that would appeal to their hands-on approach to angel investing. Especially Jesper is known for investing heavily in the team and the founder(s).

Which would -under normal circumstances - be great news. I've pretty consistently gotten good feedback from both VCs and angels. But this time I sucked in a major way. As I was pitching I knew I sucked. I felt like a dog that has been beaten to the point of scared submission. And I acted the role of a loser. Because I utterly felt like one. That meeting was an image of how I felt at the time and how my performance suffered from it.

Anyway - the list of potential investors got shorter and shorter as I got the 'thanks but no thanks' emails and calls.

Bankruptcy

We had a board meeting on May 14th 2013 where we formally decided to file for bankruptcy. I filed the paperwork, helped the lawyer, etc.

The morning after filing the papers, we all cleared out the office. Not much was said except for the occasional half-hearted joke trying to lighten the mood. We agreed to meet for a piss-up a couple of weeks later when the smoke had cleared.

It was two weeks of extremely ambivalent emotions.

On one hand I was crushed. Watching two years of my life go down the drain cannot be described to someone who hasn't felt it.

But on the other hand I felt tremendous relief. Finally, it was over. Having known where it was heading and trying to stop us falling off a cliff was unbelievably tough. Fighting with all I had knowing that our faith was in the hands of someone else and that we needed something in the proximity of a miracle. Albert Camus writes in his self-proclaimed main work, The Myth of Sisyphus, that Sisyphus is only set free as he accepts and embraces the absurdity of his destiny. I tried to do that but didn't manage.

When I had filed the papers and we were officially bankrupt and I wasn't allowed to touch anything I felt relief. Relief that my struggle was over. Relief that finally I didn't have to get up in the morning, put on a brave face and lie to everyone around me.

And I felt shame. Immense shame BECAUSE I felt relieved. I was supposed to only be devastated. I was supposed to not have given up. The common narrative of startup founders is the one in which the founder overcomes. Where he endangers the financial health of himself and his family. Where his 20 credit cards are maxed out when he finally gets funded. I felt ashamed that I had given up too soon. That I had kept my promise to my wife and not gone beyond the financial limit that we had originally agreed. Ashamed that I had only spent half our life savings on my failure and not all of it. Logically it's nonsense but the narrative that's been created around startup life - one that I have bought into as well - is one of relentless pursuit.

Sure, we had endured some hardship. Our first office space had leaky windows and no heating. Our desks were old garden furniture that I had borrowed from a friend of mine. We didn't pay any rent, instead I would clean the kitchen of the shared office space that we were in. To cut personal living cost for us all, we ate lunch leftovers from one of the more established companies in the building and in return I did the dishes for the entire building. David rented a scrappy room in Malmö and occasionally took a weekend trip to Malmö to see his girlfriend. It wasn't that we were too posh to endure hardship.

But it still felt as if we should have done more. And shame still consumes a lot of my emotional bandwidth when I think back on Admazely.

Aftermath

I guess shame was also a major driver of my behavior in the past few months. I was afraid to admit my failure to the world. Afraid to tell people that we had gone bust. Publicly admitting it was like crossing into a whole new dimension. I had a lot of psychological equity invested in my identity as a startup founder. Now, I was just a bum without a job.

My friends would tell me to give it time and think about what I wanted to do. Don't rush it, they said. Well, there were more perspectives to it than that.

First off, there was nothing else I wanted to do. Maybe another startup at some point. But right there and then the idea of taking a job was inconceivable. I waded through job sites and talked to people and I simply could not get excited about anything.

Secondly, there was the responsibility to my family. I hadn't gotten a salary for a couple of months and my wife was on maternity leave - which despite Danish welfare left us with a gross monthly household income of around 14,000 Danish kroner. In other words, we were once again digging into our savings to pay the bills. So I needed to get off my ass and generate some kind of cash flow.

Right around that time Martin Bochineck who was an angel investor and sat on the board of Admazely called me. He had had a front row seat when I lost a chunk of his money. Martin is one of the most decent and gracious people I have ever come across in my business life. I have witnessed it first hand and I have numerous accounts from others who have had similar experiences. Martin has a moral integrity that is extremely rare. I owe a lot to him as he helped me cope both during and after my failure.

He offered to pay me a salary equivalent to my CEO salary in Admazely (which admittedly was pretty negligible) to come in and "help him out with stuff for a few months" while the dust settled. I had my doubts since I simply didn't feel as if I had anything to offer at that point in time. But I eventually accepted. And that's how I've ended up at Magnetix. It's a great place with very talented people. My role here has become more permanent. I do what I did five or six years ago, handling client accounts and overseeing projects. It's familiar ground and something I know I do well. It's something I can do without investing my soul in it and still excel.

Will I be doing this the next five or ten years? Probably not. But for the time being, it's nice being around smart people doing great work. It's a privilege to be part of a very successful company. To be able to contribute at a pace that suits where I'm at emotionally.

Failure sucks monumentally. For a founder failing it needs to be ok to talk about how much it sucks. Both the small failures along the way and the huge ones that are definitive. Failure hurts emotionally. If it doesn't, you're either not really invested or simply a shallow person lacking the ability to reflect on what you're doing. My failure has cost people money, it has put people's personal relationships at risk, it has put people's financial situations at risk and it has cost a lot of my personal credibility. I don't celebrate failure, i hate failure.

I'm still recovering from my failure. Some nights, suffering from occasional insomnia, I still sit alone in the living room thinking about what I could and should have done. My failure still haunts me and I suspect it will for a long time.

If you're a founder and you find yourself in a similar situation, feel free to get in touch. Drop me an email, ping me on Twitter or give me a call. I'll be happy to spend an hour talking about how I cope (sometimes denial is an underrated survival strategy) and see if anything I've done might help you.

Original article here: http://pschlegel.tumblr.com/post/62989782523/startup-failure-how-it-feels

Startup Failure: Springpad

Springpad Says Goodbye

To the Springpad Community,

As we announced a few days ago, we are very sorry to let you know that Springpad will be shutting down on June 25th. Unfortunately, we were not able to secure additional funding or scale to become a self-sustaining business. As part of closing our business, a portion of our team is joining Google.

At this point, our priority is to help you move forward with the data you have stored in Springpad.

Today we are releasing an export tool that gives you a few options including a full Evernote migration, a viewable HTML data backup, and an importable JSON file for other services to use. Read more below about each of these options, or go to springpad.com/savemystuff to start migrating your data now. You will have until June 25th to continue to access your data and complete your migration. At that point Springpad.com will no longer be available, all online and sync features of the mobile apps will stop working, and your personal data will no longer be stored on our servers.

We understand that this transition may be difficult for many of you, and we will try to help as much as possible. If you have questions or need help, please visit the Springpad Shutdown FAQ.

Thank you to our loyal users and partners – We couldn't have made Springpad what it was without you!

Sincerely, The Springpad Team

Original article here: https://web.archive.org/web/20140528102720/https://springpad.com/blog/2014/05/springpad-says-goodbye/

Startup Failure: Exfm

Changing Tune

When we started Exfm four years ago, we had one goal. To help people discover and listen to new music on the web. What started as a Chrome extension quickly grew to a website, iPhone app, Android app, blog widget and API. Along the way we had people from over 200 countries use our apps and more than 50 artists play at our concerts in NYC.

Four years ago the music landscape was very different. Spotify and Rdio didn't exist in America. There was no Google Music or iTunes Radio. Soundcloud and Bandcamp were just getting started. Today, music flows from all places on the web. It's easy and affordable to get and it just keeps getting better.

After an amazing four years of sweat and tears, we're ever-so-reluctantly accepting the reality of sustaining the Exfm platform as it exists today. The high costs of processing millions of new songs every month while attempting to keep that data relevant and useable is monumental. The technical challenges are compounded by the litigious nature of the music industry, which means every time we have any meaningful growth, it's coupled with the immediate attention of the record labels in the form of takedowns and legal emails. Today, subscription services are gaining in popularity and enjoy the blessings of most major labels at a non trivial cost to those companies.

All this adds up to a very challenging position for a small startup with grand visions to make any real headway. The sad news is that we are no longer going to be able to keep the Exfm website and suite of apps up and running. To ease the transition to another music service, we've created an export tool so you won't lose any of your songs, as well as a new Chrome extension that can run independently of our backend.

We still want to help create the links between the disparate music blogs and all of the music sites and apps out there. Our new Chrome extension will let you scrobble songs to Last.fm from Soundcloud. You can buy Bandcamp songs you hear on Tumblr. You can discover music on a blog and instantly save it to your Rdio or Spotify playlist. It acts as the glue between these services.

The new extension is available now. If you don't have it yet, we hope you install it and give it a try. On January 15, 2014, we will completely remove the ability to love songs and follow people and your data will no longer be accessible. Our website will only offer the Chrome extension for install and our iPhone and Android apps will stop working. Our blog widget will also stop working. We urge you to grab your songs and sites before it's too late.

Once you've exported your songs you can import them into an awesome new desktop app called Tomahawk which works really well with our new Chrome extension. You call also import all your songs to Rdio using this great website - re/spin.

This isn't a full goodbye. We feel the time is right to reassess how we can continue to play a role in the new music landscape. We hope you continue to find us useful in the ever-growing world of music online. We'd love to hear back from you so please leave us comments and let us know your thoughts. Let us know what you think of the new Chrome extension.

Thanks for an amazing four years.

Best,

Dan Kantor @dankantor Marshall Jones @majman Jason Culler @jasonculler

Original article here: http://blog.ex.fm/post/70615261278/changing-tune

Startup Failure: Findory

Starting Findory: In the beginning

For the first post, I wanted to talk a bit about how I was motivated to start Findory.

The fundamental idea behind Findory is to apply personalization to information. Help people deal with information overload. Help cut through the noise and filter up the good stuff. Help people find what they need.

Personalization can do this using implicit information about your wants and needs, learning from your behavior. Personalization complements search. Search requires people to explicitly provide a query. Personalization surfaces useful information without any explicit query.

Personalization can also be used as a way to improve search. For example, if people are repeatedly refining a query (e.g. [greg linden findory], then [greg linden amazon]), they are not finding what they need. Paying attention to what they have done, especially what they have done recently, and showing different search results to different people can help surface information that otherwise might be buried deep. That is also personalization.

In addition, personalization can be used to improve advertising. Almost all advertising is useless, and annoying. Advertising does not have to be that way. Advertising can be useful information about products and services you might actually want. By paying closer attention to your interests and needs, advertising can be targeted instead of sprayed. It can be relevant and helpful, not irrelevant and wasteful. That is also personalization.

Back when I became really interested in this and wanted to work on it, I was just getting out of Stanford in 2003. Creating a company was not my first plan. I talked to Google, Yahoo, MSN, and Amazon first. The people I talked to expressed interest, but not urgency, and did not want to move as aggressively on personalization as I did.

I believed in the value of applying personalization to information. I wanted to see it happen. While I thought it was inevitable that everyone will be doing personalization of information over the next 5-10 years, I wanted to see it sooner. I started Findory to make it happen.

Starting Findory: On the cheap

When I started Findory, I was extremely sensitive to burn, probably too sensitive.

Maybe having witnessed all the dot com flameouts caused me to be overly cautious, but I wanted to be very careful about money flowing out the door. Many companies do fail early simply by running out of cash. I thought the best way to avoid this was to be careful about spending money.

Findory was built on the cheap. It is self-funded. There is no office space; the company is virtual. There is almost no marketing or advertising.

There are no salaries or benefits at Findory. For most tech startups, salaries dominate the burn rate. Even at 1/3 salary, each additional employee is about $4-6k/month (fully loaded, includes benefits, expenses, office space, etc.) At full salary, it is about $10-15k/month. Salary and benefits would have burnt through initial funding in less than a year.

By the way, it is worth noting that not drawing a salary is roughly equivalent to taking a salary but investing an equivalent amount immediately back in the startup. The latter is a lot more complicated, but both serve to fund the company using the money normally paid in salaries and benefits.

Findory uses free open source software (Apache, MySql, Linux, Perl, Berkeley DB). This allows the company to keep its costs low in its infancy. A quick note here, using Fedora Linux was a mistake. As it turns out, the upgrade path to get patches on major releases (e.g. Core 3 -> Core 4) is quite painful.

Findory uses cheap, leased, commodity servers. Findory currently runs on six servers. The cost per server is under $100/month. They are simple, low end, AMD boxes with a reasonable amount of memory and a cheap IDE disk.

Rather than pay for a news feed, Findory runs its own crawl. That allows us to customize the crawl to our needs and quickly launch new products (e.g. video, podcasts), but does take a fair amount of time to maintain and extend.

In retrospect, I think I have been too cheap. The tradeoff here is between death by burn versus death by resource starvation. While Findory has lived a long time with very low burn, it has been starved of resources, slowing growth and making it difficult to pursue many paths for expanding the products.

Starting Findory: Legal goo

Startups have to do a fair amount of legal goo. Finding a good lawyer is important not just for legal advice, but also business advice and contacts.

It is hard to find good lawyers. If you can, try to find someone with substantial startup business experience in your sector (e.g. internet startups). Talking with someone who understands your business, is excited about it, and can offer parallels to others is extremely valuable.

Unfortunately, legal work is extremely expensive. $400+/hour is not uncommon. While many law firms will delay billing for small startups and may even write off some of the work, lawyers can do major damage to the burn rate.

Even simple things like assisting with negotiating an NDA can cost a few thousand dollars. That can be annoying. Folks at large companies you might be talking to think of legal work as costless or, at least, cost is unimportant. So, biz dev people at big companies are happy to play games all day long with you with legal agreements.

But, in these discussions, each time your little startup needs to turn to your legal team, you rip through the nest egg. More substantial legal issues may cost enough to even threaten the life of a startup.

On the issue of intellectual property protection, Findory did seek patents, but I am not sure I would recommend this path to others. Filing was expensive ($10k+) and extremely time consuming, stealing time and money away from building products, helping customers, and innovating on the core engine. Even worse, I now fear that patents provide little protection for a small startup. Not only do they take years to issue, but also they cost at least $1M and several years to attempt to enforce; clearly that amount of time and money is impossible for a small startup.

There are a few other pieces of legal goo I will comment on briefly. Watch out for conflicts of interest with bigger clients of the firm; Findory lost its first two law firms suddenly and inconveniently when they bailed due to conflicts of interest. Choose your incorporation carefully; LLC or S Corp provides tax advantages, but you need to be C Corp to take some types of funding, and converting can be a bit of a pain. Put reasonable terms of use and privacy policy pages up on your website. Grab related domain names; I neglected to get a common misspelling for findory.com -- findery.com -- which still causes headaches sometimes. Make sure your intellectual property and ownership agreements are in place for anyone who does work for your firm, including your contractors.

Overall, looking back at my experience, I think a mistake I made in the very early days of Findory was thinking too much of law firm as a mere source of legal advice. In fact, the most valuable interactions with our lawyers were the business advice and connections they provided. Despite the cost, it is worthwhile to find one that can advise you on all aspects of your business.

I want to end by emphasizing that it is important to understand that any business discussions with any large firm are going to be costly for a startup. Not only do you have the risk of disclosing information (regardless of NDAs) to a potential competitor, but also the cost of any legal frameworks necessary for the discussions can be risky for a tiny startup to bear. Be very careful with who you decide to talk to and make sure those discussions are worth the cost.

Starting Findory: Launch early and often

When I was at Amazon, I was a strong believer in the idea of launching early and often. In our group, new features typically launched in weeks, then were modified repeatedly. Often, projects were completed in just days. Projects spanning multiple months were unusual.

I carried this philosophy over to Findory. Findory was built on rapid prototyping and early launches.

When Findory first started, the company was targeting personalized web search.

A quick prototype revealed the need for more data about user behavior. The recommendation engine reached out to ask, "What have other users done?" and always got the answer, "I don't know."

For a brief time, I looked at whether I could acquire the necessary information about user behavior. Were there old weblogs available from major search engines (without restrictions for commercial use)? Would Yahoo consider a deal to share an anonymized sample of their logs? Could I acquire logs from an ISP (as Snap.com later did)? No, no, no.

With some amount of frustration, I abandoned this web search prototype and switched to news. With news, bootstrapping is a bigger issue. Old news is no news, as they say. The advantage some may have from massive amounts of log data is diminished by the need to a good job handling the cold start problem.

I spent some time prototyping a personalized news application. I iterated. Eventually, it looked good in my tests.

I decided to launch Findory.com as a personalized news website in Jan 2004. At that point, Findory did not have its own crawl. Findory did not even have a search engine. But the personalization worked. I iterated on it rapidly over the next few months, launching new versions nearly daily, including a Findory crawl and a custom news search engine.

In June of 2004, I launched a version of Findory for weblogs. The underlying personalization engine had to be adapted for blog articles, articles that are often short, almost always opinion, and frequently crappy. Several iterations on an internal prototype tuned a version of the engine to the characteristics of weblogs. A crawl had to be created to cover the most interesting and useful weblogs. Tools were needed to process new weblogs and filter out spam weblogs (splogs).

With Findory and weblogs, I made the mistake of launching as a separate site called Blogory. It was a foolish decision and one that later would need to be reversed. Blogory is now merged into Findory.

Over the next many months, we iterated, iterated, and iterated. There were several redesigns of the website, all looking to figure out what format worked best for readers.

In May of 2005, personalized advertising launched. Findory's advertising (which is sourced from Google) is targeted by Findory not only to the content of the page, but also to the specific history of each Findory reader. Findory focuses the ads both based on what each reader is currently doing and on what that person has done in the past.

Now, in 2006, Findory has many products: news, blogs, video, podcasts, advertising, and (alpha) web search. Findory is personalizing information in many different areas.

Looking back, I think a few things are critical for a strategy of launching early and often.

First, you need developer websites, a full testing environment where programmers have a toy version of the website with which they can tinker. To iterate and test, you need to see everything working together in a sandbox environment.

Second, you need some objective test of the ideas, some measure of what is good or bad. You need some way of evaluating the prototypes even if it is not ideal.

One important point is that not all prototypes should be launched. Yes, I know, you worked hard on that thingie. But, many things should be throwaway, just experiments. At Findory, many things did not look good during testing and were abandoned. They were a path not followed. That work is still valuable, mind you. We learned what works and what does not work. That information is valuable even if the feature was discarded.

Finally, I want to point out that the strategy of launch early and often is not without cost. It is true that some people and press looked at early versions of Findory.com and saw things that were still really prototypes. Sometimes, they judged it harshly and never returned. That has real cost. The damage to the brand is not insignificant.

I remain a strong believer in launching early and often for web development. Findory is learning constantly. We get immediate feedback. We get rapid gratification. The benefits far outweigh the costs.

Starting Findory: Startups are hard

There is no doubt about it. Startups are hard. You have to do everything -- and I mean everything -- yourself.

You have to do all the technical work, including prototyping, research, coding, web development, design, data analysis, metrics, system administration, networking, hardware procurement, and database administration.

There is marketing, including advertising, press relations, viral campaigns, framing the product, and pitching to influencers.

There is business development, including handing requests for technology licensing, talking about cross marketing deals, exploring feed licensing, and investigating partnerships.

There is management, including project management, building a culture, choosing founders and advisors, hiring, mentoring, setting expectations, and building teams.

There is legal, including creation of and modifications to the corporate structure, non-disclosure agreements, licensing agreements, employee incentive packages, patents, intellectual property assignments, and frameworks for partnerships.

There is finance, including accounting, taxes, licensing, managing cash flow, business planning, and pitching potential investors.

There is customer service, including building help pages, creating customer self-service features, and handling incoming contacts with suggestions, complaints, requests for information, and requests for help.

All this for incredible risk. Often, it is your own money being poured into the company. There typically are no or limited salaries and no benefits.

That incredible risk comes with many rewards. Though unlikely, there is a large potential payoff dangling off at the horizon. Startups are building something new, the exciting tip of the cycle of creative destruction. And, running a startup, you have much freedom in what you decide to pursue.

However, that freedom can be a curse. There are so many degrees of freedom, it can be overwhelming.

What should the company do? Even more important, what should it not do? Does it have enough money to pursue X? Can a competitor better implement feature Y? What really matters to the company?

There are so many options, but so little time. You must keep moving, keep making decisions, but always be willing to stop or reverse if something looks wrong.

You learn a lot, but it is incredibly difficult. It is a remarkable challenge, an incredible experience.

Starting Findory: Talking to the press

I wanted to do a piece on talking to the press. I am a tech geek, not a marketing dweeb, so talking to the press is not my strong point. But, having been forced to do it, I think I have learned a few things.

The biggest thing I have learned is to focus on what reporters want. It seems to me that reporters are looking to write a solid, interesting story, supported by good sourcing and strong quotes. A typical story seems to consist a few paragraphs of explanation followed by some quotes from someone that express an opinion on the reporter's explanation.

My strategy is to only try to make a few key points. I try to say a few things that describe my impression of the overall situation, intending that piece to help with the explanation that is in the reporter's own words.

Then, I try to give just two or three pieces of opinion, my view on the situation. At this point, I try to talk slowly and carefully, keeping in mind that the report probably is trying to transcribe an exact quote. Sometimes, I repeat the quote or something fairly similar to the quote.

One other thing that I have found important is that reporters do not always ask the questions you think you they should. That is okay. You can usually take a question that you may feel is poorly chosen or misdirected and simply answer a variant on it that is the question you thought should have been asked. That variant is just as likely to make it into a quote

as a more directed answer and may be more appropriate and more useful to the reporter.

Finally, I have found that most press coverage of tiny, little startups like Findory will be positive. After all, if they were going to write something negative, they would not bother writing about Findory at all. Yes, being ignored is an issue, but it is good to know that, when you are talking to the press, if you get any coverage, you are likely to get fairly positive coverage.

Maybe it is working on a news site, but I tend to have a lot of sympathy for the press. I think reporters are struggling with a difficult job and try to help them as much as I can. However, helping reporters is not a selfless act. The more helpful I am, the more likely it is that I get in the story, and any mention of Findory may benefit the company.

Starting Findory: Customer feedback

Findory customer service is light, averaging a couple e-mails per day. They are mostly suggestions, a few requests for help, and a few complaints.

Of the few complaints, the most common are various forms of rants about either liberal bias or conservative bias. Partially this is due to our political climate right now where any site at all related to news is bombarded by absurdity from the extreme fringes.

However, accusations of bias also may be due to Findory explicitly trying to not pigeonhole people. The idea of personalized news has been around for a decade or so. One common criticism of the idea is that personalization may pigeonhole people, showing them only what they want to see. On Findory, opinion articles are not selected based on a particular view, with the result that people are exposed to viewpoints they might prefer to ignore.

It is an interesting question whether Findory should have given people what they wanted -- let people put on their blinders and pigeonhole themselves -- or if it is doing the right thing for the long-term by helping people discover a breadth of information and viewpoints.

On the requests for help, the most common is somewhat amusing. People often expect Findory to be harder to use than it is. It surprises me, but some write in and ask, "How do I set it up? What do I do?"

Perhaps Findory is too simple. "Just read articles! That's it," I often say. "Findory learns from what you do." Everyone has been trained to expect sites to be more difficult -- lengthy registration and configuration, for example -- and it appears it can be confusing when all those barriers are removed.

On the suggestions, the most common are requests for a feed reader (which we did as Findory Favorites), interest in rating articles and sources (which we have prototypes a few times but never launched because it seems to change the focus away from reading), a desire to see news photos inline on the page (potentially costly, but something we are exploring), extending the crawl (always working on it), and support for non-English languages (prohibitively expensive due to the changes required in the recommendation engine).

Customer feedback is useful. Many Findory features have been implemented as a direct result of suggestions from Findory readers.

However, just listening to customers is not enough. Customers will tend to suggest iterative improvements and request more features. Customers will not offer ideas for big jumps, big ideas, or major new products. Customers will not balance requests for new features against simplicity and usability.

When looking at customer feedback, I find it is important to look beyond the words to try to divine the intent. The best solution may be something completely different than what was suggested.

Starting Findory: Marketing

Ah, marketing. Is there anything that techies like less?

It is obviously naively idealistic, but I think we geeks wish marketing was unnecessary. Wouldn't it be nice if people could easily and freely get the information they need to make informed decisions?

Sadly, information is costly, and the time spent analyzing information even more so. People generally do use advertisements to discover new products and rely on shortcuts such as brand reputation as part of their decision-making.

As much as we might hate it, marketing is important.

Marketing also is absurdly expensive. It is mostly out of reach for a self-funded startup. Though I recognized the need, Findory.com did almost no traditional marketing.

There were limited experiments with some advertising. For the most part, these tests showed the advertising spend to be relatively ineffective. The customer acquisition costs came out to a few dollars, cheap compared to what many are willing to pay, but more than a self-funded startup reasonably could afford.

Even without substantial advertising, for the first two years, Findory grew at about 100% per quarter. Most of this was from word of mouth and viral marketing features.

Findory tried to accelerate word of mouth by focusing marketing time and effort on PR with reporters and bloggers, sharing data through RSS feeds and APIs, and providing Findory content to websites and weblogs with Findory Inline.

Findory's growth has stalled lately, casting some doubt on the strategy of pursuing word of mouth and viral marketing alone. Again, the question comes up over whether to spend time and treasure on non-traditional or traditional marketing.

Starting Findory: The team

Building a strong team around a startup is important. No doubt about it, I was slow to build a team for Findory.

Findory was started with a very modest amount of capital, probably too small. The on the cheap strategy meant Findory could not pay salaries or benefits. That made hiring and building a team difficult.

I did not want to bring people on part-time who were not dedicated to success of company, but I could not offer what people need -- at least a modest salary -- to come on full time.

In retrospect, I should have increased the initial capital invested, paid small salaries, and brought more people on the team. I was too concerned about the burn rate and not concerned enough about getting the right people on board quickly.

I also should have brought on advisors and formed an advisory board. The assistance and connections would have been valuable. Worries about loss of independence and dilution of ownership should have taken a back seat to getting the advice I needed.

Finally, I should have brought an angel on the team. As I learned with our lawyers, the advice and network I would have gained from a dedicated angel would have been as valuable as the funding. A modest amount of additional funding also would have been sufficient to pay 1/3 salaries to a couple employees, easing the build out of the critical early team.

It is ironic. I started Findory because I was passionate about an idea and loved the speed of execution I gained from my independence. In my excitement, I forgot the importance of sharing that passion with others and depending on their advice.

Starting Findory: Infrastructure and scaling

From the beginning of Findory, I was obsessed with performance and scaling.

The problem with personalization is that that it breaks the most common strategy for scaling, caching. When every visitor is seeing a different page, there are much fewer good opportunities to cache.

No longer can you just grab the page you served the last guy and serve it up again. With personalization, every time someone asks for a page, you have to serve it up fresh.

But you can't just serve up any old content fresh. With personalization, when a visitor asks for a web page, first you need

to ask, who is this person and what do they like? Then, you need to ask, what do I have that they might like?

So, when someone comes to your personalized site, you need to load everything you need to know about them, find all the content that that person might like, rank and layout that content, and serve up a pipin' hot page. All while the customer is waiting.

Findory works hard to do all that quickly, almost always in well under 100ms. Time is money, after all, both in terms of customer satisfaction and the number of servers Findory has to pay for.

The way Findory does this is that it pre-computes as much of the expensive personalization as it can. Much of the task of matching interests to content is moved to an offline batch process. The online task of personalization, the part while the user is waiting, is reduced to a few thousand data lookups.

Even a few thousand database accesses could be prohibitive given the time constraints. However, much of the content and pre-computed data is effectively read-only data. Findory replicates the read-only data out to its webservers, making these thousands of lookups lightning fast local accesses.

Read-write data, such as each reader's history on Findory, is in MySQL. MyISAM works well for this task since the data is not critical and speed is more important than transaction support.

The read-write user data in MySQL can be partitioned by user id, making the database trivially scalable. The online personalization task scales independently of the number of Findory users. Only the offline batch process faced any issue of scaling as Findory grew, but that batch process can be done in parallel.

In the end, it is blazingly fast. Readers receive fully personalized pages in under 100ms. As they read new articles, the page changes immediately, no delay. It all just works.

Even so, I wonder if I have been too focused on scaling and performance. For example, there have been some features in the crawl, search engine, history, API, and Findory Favorites that were not implemented because of the concern about how they might scale. That may have been foolish.

The architecture, the software, the hardware cluster, these are just tools. They serve a purpose, to help users, and have little value on their own. A company should focus on users first and infrastructure second. Despite the success in the design of the core personalization engine, perhaps I was too focused on keeping performance high and avoiding scaling traps when I should have been giving readers new features they wanted.

Starting Findory: Hardware go boom

Computers go down, a lot more often than we might like.

For most of Findory's four years, it ran on six servers. In that time, those servers had one drive failure, one bad memory chip, and four power supplies fail.

Of these, only the two power supply failures caused outages on the site, one of one hour and one a painful eight hour outage. There were a few other short outages due to network problems in the data center, typically problems that took the entire data center offline temporarily.

Findory's six servers were all cheap commodity Linux boxes, typically a single core low-end AMD processors, 1G of RAM, and a single IDE disk. Findory was cheap, cheap, cheap.

The reliability of Findory over its lifetime was perfectly acceptable, even quite good compared to other little startups with similar levels of traffic, but I think it is interesting to think about what may have been able to prevent the outages without considerable expense.

Surprisingly, RAID disks would not have helped much, though they would have made it easier to recover from the one drive failure that did occur on a backend machine. Better redundancy on the network may have helped, but would have been expensive. Splitting servers and replicating across data centers may have helped, but would have been both expensive and complicated for a site of Findory's size and resources.

Probably the biggest issue was that Findory did not have a hot standby running on its small database. A hot standby database would have avoided both the one hour outage and the eight hour outage. Those outages were caused by losing the first, then a second power supply on the database machine.

Looking back at all of these, I think it was particularly silly not to have the database hot standby. The cost of that would have been minimal and, not only would it have avoided the outage, but it would have reduced the risk of data loss by having a constant database backup. I almost added the hot standby many times, but kept holding off on it. While I may have mostly gotten away with it, it clearly was a mistake.

Starting Findory: Funding

I had the wrong strategy with trying to get funding for Findory. I pursued VCs instead of angels.

I should have realized, VCs would never fund Findory. From their point of view, there was just too much risk.

Findory was lead by a first time entrepreneur and was missing critical expertise in building a company. Findory lacked a full team; in particular, there was no finance, marketing, or business development talent. And, Findory had an unproven technology -- a specific technique for personalizing news, search, and advertising -- that is difficult for a VC firm to evaluate.

Had any one of these three been different, there might have been a better chance. A proven founder may be able to get funding with nothing more than an idea on a napkin. A strong team ready to go is an attractive investment for a VC. A proven technology that just needs the wind of capital behind it is an easy bet. But, with all three missing, there was never much of a chance.

Perhaps in the frothier SF Bay Area, this might have been different. Perhaps someone might have taken a gamble. In Seattle, there are only a half a dozen VCs doing substantial internet investing; most firms out of Seattle will not fund without a local firm joining in the round.

I did pitch dozens of venture capitalists in and outside of Seattle on Findory. It was a fascinating experience, educational, exciting, and often humbling. But, ultimately, I have to say it was a wasteful distraction from building things to help Findory readers.

I should have focused on angels. Not only would angels have brought the right amount of cash for short-term needs, but also the addition of experienced angels to the board would have provided valuable advice and connections. It would have been just what Findory needed.

I had thought that, with how far I had bootstrapped Findory, that I could skip a step and jump directly to VCs. That was foolish. Findory remained a long way from the point where the momentum would be so obvious that it would overwhelm the other concerns.

Instead of looking to VCs, I should have looked for an experienced angel, someone passionate about personalizing information, who would have given the startup the advice it needed while enjoying the experience of joining with us to grow the company.

Please see also Paul Graham's article, "A Hacker's Guide to Investors".

Please see also Paul Graham's article, "Why There Aren't More Googles", especially his discussion of VCs as tending to be surprisingly "conservative" and "driven by consensus".

Starting Findory: Acqusition talks

At various points when I was running Findory, I approached or was approached by other firms about acquisition. For the most part, these talks went well, but, as with many experiences at Findory, my initial expectations proved naive. It gave me much to contemplate, both for what startups should think about when entering these talks and changes bigger companies might consider in how they do acquisitions.

For a startup, acquisition talks can be a major distraction. It is time away from building features for customers. It creates

legal bills that increase burn rate. It distracts the startup with nervous flutters over uncertainty over the future, potential distant payouts, and the complexity of a move.

Acquisition talks also can be dangerous for a startup. Some companies might start due diligence, extract all the information they can, then decided to try to build themselves.

There is some disagreement on this last point. For example, Y-Combinator's Paul Graham wrote, "What protects little companies from being copied by bigger competitors is ... the thousand little things the big company will get wrong if they try." Paul is claiming that big companies have such poor ability to execute that the danger of telling them everything is low.

However, big companies systematically underestimate the risk of failure and cost of increased time to market. For internal teams, which often already are jealous of the supposedly greener grass of the startup life, the perceived fun of trying to build it themselves means they are unreasonably likely to try to do so. Paul is right that a big company likely will get it wrong when they try, but they also are likely to try, which means the startup got nothing from their talks but a distraction.

There are other things to watch out for in acquisition talks. At big companies, acquisitions of small startups often are channeled into the same slow, bureaucratic process as an acquisition of a 300-person company. Individual incentives at large firms often reward lack of failure more than success, creating a bias toward doing nothing over doing something. In fact, acquiring companies usually feel little sense of urgency until an executive is spooked by an immediate competitive threat, at which point they panic like a wounded beast, suddenly motivated by fear.

Looking back, my biggest surprise was that companies show much less interest than I expected in seeking out very small, early-stage companies to acquire.

As Paul Graham argued in his essay, "Hiring is obsolete", for the cost of what looks like a large starting bonus, the companies get experience, passion, and proven ability to deliver. By acquiring early, there is only talent and technology. There is no overhead, no markups for financiers, and no investment in building a brand.

Moreover, as the business literature shows, it is the small acquisitions that usually bring value to a company and the large acquisitions that destroy value. On average, companies should prefer doing 100 $2-5M acquisitions over one large $200-500M one, but business development groups at large companies are not set up that way.

There is a missed opportunity here. Bigger companies could treat startups like external R&D, letting those that fail fail at no cost, scooping up the talent that demonstrates ingenuity, passion, and ability to execute. It would be a different way of doing acquisitions, one that looks more like hiring than a merging of equals, but also one that is likely to yield much better results.

Starting Findory: The end

This is the end of my Starting Findory series.

Findory was my first startup and a nearly five year effort. Its goal of personalizing information was almost laughably ambitious, a joy to pursue, and I learned much.

I learned that a cheap is good, but too cheap is bad. It does little good to avoid burning too fast only to starve yourself of what you need.

I re-learned the importance of a team, one that balances the weaknesses of some with the strengths of another. As fun as learning new things might be, trying to do too much yourself costs the startup too much time in silly errors born of inexperience.

I learned the necessity of good advisors, especially angels and lawyers. A startup needs people who can provide expertise, credibility, and connections. You need advocates to help you.

Original article here: http://glinden.blogspot.ca/2009/10/starting-findory-end.html

Startup Failure: Findit

FindIt is shutting down – Thank you for all the support!

As of today, we are shutting down the FindIt application and focusing our team and resources on an entirely new product. This will not impact your files or emails in any way. Our servers will be shut down, and the FindIt application will no longer be functional.

This is a bittersweet moment for our team. Starting a company and trying to change the world is no easy task. We had a vision to empower people with simple and intuitive search on their phone. Many of our users told us incredible stories about FindIt helping them in moments of need, but in the process we learned that the majority of our users did not need FindIt often enough to justify our continued time and effort on this problem. Thank you for continuing to inspire and motivate us.

While building FindIt we faced the challenge every business faces, how to find new customers. We discovered the power of Facebook advertising in finding new users, and we developed expertise in creating, managing, and optimizing Facebook ads. As we spoke with other entrepreneurs and business owners we learned that most do not understand the potential Facebook advertising has to help grow their businesses and they have no idea how to get started. We knew we could help.

In early 2014 we began building a solution to this problem, and we already have paying customers who love our approach. If you own a business or work in a business that needs help with Facebook advertising, check out what we are working on now at adup.co

We are truly fortunate to have the support of amazing investors and a fantastic team. We continue to be driven by our passion for simplifying technology to empower people and make their lives better. We are excited to redirect this passion to help entrepreneurs grow their businesses.

All the best from the FindIt team, and may you always find what you are looking for in life

Levi, Alex, Ben, & Brandon

Original article here: https://web.archive.org/web/20140712084626/http://www.getfindit.com/findit-is-shutting-down-thank-you-for-all-the-support-2/

Startup Failure: MyFavorites

The little startup that couldn't (a postmortem of MyFavorites)

I'm officially putting MyFavorites behind me. The problem I was trying to solve, isn't one of those problems you think you have… and I think those types of solutions/startups are a bit tougher. I already outlined my own problems with running a startup in a prior post about MyFavorites ("Repeat after me, I will not do another startup as a non-technical founder unless…").

I'm writing this post with the intentions of telling you about the idea, sharing the debates that occurred with how the app should operate and be used, share mockups with you, share documents/spreadsheets of tasks/bugs/features/timelines, and simply give a glimpse into this failure. It was a failure because I called quits on it — I ultimately couldn't keep funding it myself and the team was losing interest in working on this app that I kept going back-and-forth on how the user experience should play out — when we hadn't even had any users using it yet.

THE IDEA

My initial pitch for MyFavorites was "Show + Tell Your Favorites". That pitch eventually became: "The Like button for everything." Nick came up with that one and it was solid gold — would have made a great TechCrunch post title… we discussed possibly using it as our tagline, knowing it might be controversial and Facebook might sue — but any press is good press :) We weren't going to do that though. [Quick shout-out to Buffalo-native Chris Sacca, whose portfolio list has a one-liner that sums up every single company. If you can't do that for your startup, then you're likely too broad and not simple enough. Remember, if you can't explain your solution to your Mom so she can understand it, then most people won't understand it and they especially can't explain it to their friends]

The problem we tried to solve — wouldn't it be great to know the favorite books of your friends (and celebs), so you know what book you should be buying to read this weekend? Facebook has interests, but have you ever updated those since you signed up for Facebook? Wouldn't it be great to know your friend's favorite … anythings? You'd be looking at a feed of just favorites — blog posts, beers, sneakers, drinks at Starbucks, things to do in Baltimore, apps for iPhone, tree cutting services locally, meals at a restaurant, etc.

Wouldn't it be cool to see all of Ashton's favorites? Or Britney Spears, or any celebrity?

We were tackling the "interest graph". You can find lots of my notes and findings at the MyFavorites blog.

I'll tell you my grandiose billion-dollar idea for MyFavorites — imagine 50k people that took pictures of themselves with their love for Starbucks (or any brand). Sure, Starbucks currently can show a Facebook widget that shows profile pics of their Facebook fans… but imagine just replacing their homepage with 50,000 people showing their absolute love for Starbucks? That'd be amazing — Starbucks doesn't even need to say anything about their products, because here's people that vouch for us. Imagine Gary Vaynerchuk when selling his next book to just show tons of pictures of people with his last book, showing creatively how much they love garyvee?

With all of these pictures, a new ad network could be created — per the one I wrote about back in 2007 ("Ads with Personal Endorsements") — this is the future, there's no doubt about that. The problem is how do you get people to take pics of themselves with brands/products that they absolutely love and personally vouch for?

In the feed, our plan was to always have a #dickbar at the top of the screen. That would be asking the user for their favorites. At first, these would all be some default categories that we ask everyone, but as the user gets friends and follows people, anytime those users are asking people for favorites in a category, that user's profile pic and name would show in the #dickbar with category they want favorites for. Meanwhile, there'd be sponsored favorite questions — such as, if you favorited Doritos, then Doritos could ask you "What's your favorite Doritos flavor?", or if you favorited Starbucks, it could be asked "What's your favorite drink at Starbucks?", etc. It just keeps going and going. [Here's a spreadsheet of some more sub-question examples]

As a user favorites something, there would be sub-questions to those categories. If you favorited 'NFL', then we could ask the question, "What's your favorite NFL team?". The plan was to allow multiple favorites for any category, for every user…

so that everytime you were drinking a beer you loved, we wanted you to whip out your phone and favorite it. Then you'd have a ranking of your favorite beers — with # of times you favorited each one. You'd also be able to see the favorite beer tally of all your friends combined, or individually… so then you would know the most popular beer amongst your friends. With us knowing where you live and where you're favoriting this stuff, we could also figure out the most popular ANYTHING in every city — that's powerful. We could basically show a map of the world and show Gibson vs Fender and which guitar brand is the leader in every city, every state, and every country.

LOGO

I used 99designs. I know that basically every designer out there loathes this service and anyone that uses it, but I beg to differ. If you're awesome, you aren't going to be replaced by 99designs. I think it's a great opportunity for college kids and international designers that don't have access to design work. It's a great platform for learning as a designer — understanding a client's needs/requirements/vision and trying to design for that. The designer gets feedback from not only the client, but other designers — and can see feedback on other designer's work by the client [and others]. As a designer you need to go into knowing you're basically paying for education — you pay by doing work for free and likely not going to get paid for that work. You go to school and pay for that education. Education and experience costs money and time. 99designs is disruptive and I understand the controversy. Various designers in the world will continue using it and so will various people — but it's not going to become the defacto for the entire design world.

With the MyFavorites logo, I wanted something that could eventually be placed on blog posts like the Twitter 'tweet' button and the Facebook 'like' button.

FAVORITING PROCESS

We went back and forth on this so many times I can't count them all. Granted, I was the one changing my mind a lot. We initially started with a sentence strategy — you would say "My Favorite beer is Duvel" — so basically "My Favorite (category) is (item)". The problem you run into is pluralization — is vs are. I can't even speak to all the issues with this, but our language is a mess :) [Here's a screenshot of an old sentence structure process for our app]

The other problem with the sentence structure is that it seems a bit lengthy. We all went to SXSW 2011 and were almost ready to launch the app (this was after weeks of tons of late nights trying to cram this app to be ready) … and then we all saw Pete Cashmore's interview of Dennis Crowley … where Dens talks about making the checkin process as easy as possible.. and I realized we needed to simplify our favoriting process. Nick had mentioned many weeks prior that we should start the process by showing like 10 default categories, with the ability for user to choose 'other' and input their own category. This would help guide our users and make the process slicker. So that was our final strategy on this focal UX point of the app.

We also originally allowed the user to input multiple categories for a favorite — this seemed to like it would make the user have to "think" too much. We found ourselves wondering what categories/tags to use.

Our other strategy was to focus on the ability for users to add tips to Foursquare. Initially we thought that when someone favorites something, they'd be able to checkin at Foursquare, post it to Twitter and Facebook. That doesn't really make sense though — I use the Foursquare app — I opened that as soon as I walk into a venue — that's when I think to use that app. The idea with MyFavorites is to use it when you are loving something — so likely, you're already checked-into a Foursquare venue, and so this would be an ability to add more tips into Foursquare…. which hasn't been done yet. No other apps are focused on getting more tips into Foursquare — and to me, tips are the most valuable aspect of Foursquare.

The big problem we just never knew how to answer until the app started getting used, was we didn't know what to give the user after they favorited something. Foursquare gives points, random badges… should MyFavorites being doing game mechanics like this? Leaderboard seemed pertinent to us — but specifically showing the user what other favorites exist in the category they just favorited in — and where their current favorite item ranks in relation to other items they have favorited in that same category (i.e. 'Pabst Blue Ribbon is your 3rd favorite beer!'). [Here's an old leaderboard mockup and another one specific to a user] [Here's a spreadsheet of tons of possible messages we could have delivered to the user after they favorited something]

MOBILE APP vs WEB APP

Initially we were building both an iPhone & Android app (after establishing our dev platform as Titanium Appcelerator), as well as a website where you could favorite things as well. It was all too much. Even with Titanium's ability to "write once, and push out an iPhone and Android app" — that's false; it takes a lot of work to manipulate features for iPhone and Android — there's no scroll wheel function in Android; there's no menu button on iPhone as there is on Android.

Having a web app being created at the same time was ridiculous too — especially since we still hadn't nailed down the favoriting process or tried it with any users. I was blowing cash — at a ridiculous pace. I had 7 guys working on this thing at once, as we were hustling for SXSW launch deadline. We decided to focus on the iPhone app, which sucked for me and Dan the backend programmer, because we both couldn't even use the app — we both have Droid X phones.

Focus on one platform. Get it out there — let people use it — nail down the UX with user input.

SOCIAL INTERACTIONS

I looked at Twitter, Facebook and Foursquare — then wrote a post called the "Metrics of me, me, me!" — basically the conscious and subconscious interactions/metrics we make with these services. I also blogged about the "usage psychology" of those services to better understand how we could hook MyFavorites users. Our social interactions were the following:

- favorite [default action; like a tweet or checkin]
- 'me too' [trying to get users to add more favorites easier]
- 'reply' [ability for users to reply to someone's favorite with their own favorite -- i.e. your favorite car is a Nissan Altima, well mine is a Mattel. We found a lot of humor came from this action]
- ask for favorites [ability for user to ask friends for their favorites in a category -- i.e. if I were going to Ireland for vacation, what are things to do there?] [users would simply 'reply' with their favorite] *

API

Eventually we wanted to open an API that allowed you to have favorites auto-imported — such as Twitter favorites, who uses that feature on Twitter anyway? Our site would give you a reason to use favoriting on Twitter. For YouTube, we'd auto-import stuff you favorite in there. Lots of apps out there have a like or favorite ability, and it seems like an opportunity exists for aggregating all of that.

MOCKUPS OF FUNCTIONALITY

These obviously took tons of hours of our lives nailing these. We looked at lots of apps for help/ideas.

Question Process:

This was up for serious debate all the time. Basically, I wanted this. I wanted to be able to ask my friends what their favorite things to do in Ireland were. Or Hawaii, where I'm going for my honeymoon. Or I am looking for someone to take down this tree in my backyard — Yelp sucks in Buffalo, so how can I find a friend that can recommend a tree cutting service? So the 'where' was a big issue — I didn't want just a 'tree cutting service', I wanted a 'tree cutting service in Buffalo, NY, USA'. So we had this 'Where?' optional field, which then would either take a Foursquare venue, or you could select a city or country from a DB we had. It was a complete mess — it was confusing to users [especially when we had this on the Favoriting process] — and the rest of my team didn't think we needed this functionality at all anyhow.

Favorite page on web — we ultimately decided to do the Instagram thing, which was have a non-interactive website initially and simply have a webpage for every favorite that occurs by users, so that these could be shared. I think this minimal design was great and we planned to build out feed/category/profile pages on the web this way too.

Meanwhile at SXSW is when I first learned that Kevin Rose was gathering some troops (including Daniel Burka, whose design work I have been following for years in awe) to create Milk Inc., which was going to be an incubator of app ideas. One of those app ideas that possibly was going to be their focus (OINK), was dabbling in similar territory of MyFavorites. We had about 6 months on them, but still — that's some competition.

THE END

I still want to use an app like MyFavorites — and I hope OINK can nail it. Ultimately, I wasn't the guy to push this idea through. Being a non-technical founder, I just can't throw money at this thing in the hopes of nailing it. I believe we were definitely at a point where we could have raised some funding around SXSW timeframe — we had the team, the focus, and an app that was working … but ultimately when we came back from SXSW, we all started losing interest, the team was all wondering where this was eventually going, and I was wondering if I even wanted to run a startup, have investors, have the responsibility of employees and answering to a board of investors, etc.

Moving forward I'm looking to help the OINK guys or anyone working on this problem in any way I can. There's a solid 6 months of problems, solutions, strategy, userflow, monetization, and everything else under the sun… that went down. And there's still a massive opportunity out there to nail.

Original article here: http://www.stevepoland.com/the-little-startup-that-couldnt-a-postmortem-of-myfavorites/

Startup Failure: Inq Mobile

Inq Mobile, One Of The First Facebook Phone Makers, Shuts Down

Inq Mobile, one of the first companies to build a Facebook phone, announced that it has shut down with a message on its site (h/t Android Police). In a statement, the U.K.-based, Hutchison Whampoa-backed company said:

"Inq has been a really exciting business over the last few years and whilst there have been significant successes, the technology that's been borne out of that work has been identified to have greater application within the wider Hutchison group. Consequently, we've taken the hard decision to close the Inq business down.

We want to thank all the customers and employees that have engaged with Inq's products and experiences over the years.

Inq's software services Material and SO.HO will be closed down at the end of January. For more information please go to http://www.inqmobile.com."

Inq, which was founded in 2008 and pivoted a year ago to focus on mobile software, said it will no longer update Material and SO.HO, its apps. Material, a news reader, released its final editions on Jan. 28, while social media aggregator SO.HO will not be updated after today, though it will continue to function. Support pages for the Cloud Touch smartphone and Inq's featurephones remain on its site.

The timing of Inq's closure and Material's shutdown is interesting because several of tech's largest companies have recently started to offer their own news apps and tools. These include Yahoo's News Digest; Twitter and CNN's Dataminr; and Paper by Facebook, which will launch next month.

Inq Mobile began as a maker of low-priced Android smartphones. It was one of the first companies that collaborated with Facebook to create a social smartphone in 2011, around the same time HTC and the social network struck the partnership that yielded the Salsa and ChaCha.

Inq's Cloud Touch, which was released exclusively in the UK three years ago, had a custom Facebook wrapper built on top of Android, and an early version of SwiftKey. Though cheaply priced (starting at $50 with a subsidized contract), the Cloud Touch couldn't compete with Samsung's rapid takeover of the Android market. The company pivoted and started developing mobile apps one year ago.

Material, which TechCrunch covered when it launched its iOS version in August, was a social magazine app that used Inq's "interest extraction engine" to look at the Facebook and Twitter accounts of users and figure out what kind of articles they wanted to see. Content was delivered in two daily editions.

At its launch, Material already had strong competition from popular social news readers like Flipboard, Zite, and Pulse.

Inq CEO and co-founder Ken Johnstone told TechCrunch at the time that Material differentiated from other news readers by offering an easier set-up than its rivals because all users needed to do to power Material's algorithms was connect their Facebook or Twitter accounts.

"For somebody who has invested a lot of time in Twitter and Facebook anyway, this is about getting a return on that investment," Johnstone told TechCrunch's Natasha Lomas.

Yahoo, Twitter, and Facebook's news aggregation products all feature some human curation, but, like Material, they also rely heavily on algorithms to customize content for each user. Inq had planned to monetize Material by harvesting enough data to build an advertising business, but its failure to do may be a cautionary tale for other developers of news readers, even as they continue to rethink how content is organized.

Though algorithms are necessary if a news aggregator wants to scale up (and collect enough data to be profitable), they still can't replace the discernment of a human editor. Like Feedly, Pulse, and Zite, Material's customized content stream suffered from problems like miscategorized stories, irrelevant content, and "the overall feeling you get from flicking through an edition is not a cohesive, editorially unified whole, but an algorithmically generated bunch of mostly random

stories with (at best) a few loose, overlapping themes," as Natasha put it.

Original article here: http://techcrunch.com/2014/01/30/inq-mobile/

Startup Failure: Outbox

Outbox is Shutting Down—A Note of Gratitude

Dear loyal customers and Outbox supporters:

We announce today that we are ending the mail service, shutting down the Outbox brand, and focusing our team and resources on a totally new product. In a related post we explain what this means for our current, loyal customers.

This is bittersweet for our team, since we have poured so much into creating our product and serving our customers. Yet this announcement marks the end of a chapter for us—not the end of the story.

Over the past two years, we have been humbled by the support of our investors and customers, who each took a leap of faith to join us in this exceptionally audacious venture. We have also been humbled by the incredible personal and financial sacrifices our team members have made along the way.

The Beginning

We set out to redefine a long cherished but broken medium of communication: postal mail. We did so during a tumultuous period in the history of the United States Postal Service (USPS), which has experienced declining mail volume and staggering deficits for the past five years.

As former staffers on Capitol Hill, we share a passion for tackling huge public problems, but aim to do so with private solutions. We knew that the USPS would not be able to work out its own problems so, perhaps naively, we hoped to partner with USPS to provide an alternative to the physical delivery of postal mail to a subset of users, hoping this would spur further innovation and cost savings.

Although an early test with the USPS that let users redirect their mail to us showed signs of success and operational simplicity, an interview by CNBC triggered a request from the Postmaster General himself to meet in Washington, DC. In one of the most surreal moments of our lives (listen to it in an NPR interview), we had our very own Mr. Smith Goes to Washington encounter where the senior leadership of USPS made it clear that they would never participate in any project that would limit junk mail and that they were immediately shutting down our partnership. This 30-minute meeting was the end of our business model.

The Reimagining of Outbox

After countless hours applying Clay Christensen's business model theories to our situation, we came to view our failed partnership with USPS as a David and Goliath moment: we believed our seeming disadvantage would become our greatest strength. Turning our original vision on its head, we reimagined our service as not merely playing in someone else's value channel, but as a new type of last-mile delivery channel all together: one subsidized by our users in return for collecting and electronically delivering their postal mail. If we could simply break even on the mail business, we would have built a valuable last mile network able to be monetized in many ways.

To pull this off, we built a world-class team of engineers, designers, marketers, and operations specialists in Austin and San Francisco. Funding our efforts were some of the most celebrated investors of our generation: Mike Maples at Floodgate and Peter Thiel and Brian Singerman at Founders Fund, as well as a groundswell of investors via AngelList. Together, we made a product that was as beautiful as it was complex, and overcame nearly every obstacle in our path.

We created our own dynamic logistics software, developed a legal framework to open users' mail, built industrial-grade scanning machines for 1/100th of the market price, developed specialized OCR to allow customers to unsubscribe from postal mail, built and attached to our cars 5-foot mailbox flags that withstand 70 mph highway speeds, laser cut wood blocks to build mail slot solutions, and created a novel system of key decoding via photograph that inspired the creation of one startup all on its own. All this was simply the backend of our service, and our iPhone and other apps won awards for their design and elegance.

In the end, we serviced a little over 2,000 individual customers, had 25,000 people waiting around the country on our

waiting list, unsubscribed our customers from over 1 million senders of mail, scanned over 1.5 million pages, and delivered over 250,000 requested mail packages. We also recycled approximately 30 tons of paper, enough to cover 86 football fields.

Outbox was buzzing. It seemed as though everyone knew something about our little company, had seen one of our red-flagged mailbox cars, or had stumbled upon a news story about us. CNN praised us, Jay Leno mocked us, and Pee Wee Herman called us "the future." We tested our anecdotal suspicions with a nationwide survey, and found that Outbox had an unaided brand awareness of 10.1 percent - even though we serviced a mere 2,000 customers in two relatively small markets.

Numbers Don't Lie

After raising $5m in June of last year, we set out to onboard the 4,000 individuals we had amassed on our central-San Francisco waitlist. We projected converting a large percentage of these individuals, and planned to scale our marketing efforts at a projected cost of $20 per acquisition.

However, after an extensive email marketing campaign to our waitlist, total yield from the waitlist was under 10 percent. And as we started marketing outside of this network, we had difficulty finding a repeatable and scalable acquisition channel. Across all of our efforts, our acquisition numbers were over $50 per lead.

As our marketing efforts lagged behind schedule, our density numbers remained consistently flat, causing us to spend about double our projected cost to service each customer. Even our most dense routes cost us approximately 20 percent more than our break-even target.

After several months of testing and refining, we reasonably concluded that we were executing well and collecting good data—it told us that there wasn't enough demand to support the cost model. Our monthly operating deficits were too high, and even though we continued to get better at acquisition, each small success actually saw our cash curve decline further because our density remained flat. For longer than we would be willing to tolerate, we would lose money for each additional customer we gained. Despite the massive interest in our company, we learned that the product we built did not find fit in the market we targeted.

Finding serenity in knowing when to stop For startups, it's difficult to know when to throw in the towel. Indeed, the main strategy for most of the life of a startup is overcoming impossible odds, and we built a team that did that over and over again.

This final challenge—product market fit—is one we ran after with characteristic zeal. Amidst these struggles we were reminded of the serenity prayer written by one of our favorite authors, Reinhold Niebuhr:

Grant me the serenity to accept the things I cannot change, The courage to change the things I can, And wisdom to know the difference.

Facing situation after situation we kept the courage to change them; in these final few months, we were granted the serenity to know this situation is one we cannot change.

Our Unpostmen

Not least of our accomplishments was the recruitment of an amazing operations team. Their tireless work ethic led to such stellar customer service that our users would, on occasion, pee their pants. The hardest reality in our transition is saying goodbye to our eighteen Unpostmen: they have been the frontlines of challenging logistics and extreme customer service. If you are hiring and looking for loyal, hard working operations experts, please email us here.

Our Future There are numerous small pivots we could make as an alternative to our mail service, and we have tested many other applications that could be layered on top of the logistic network we already created. Yet each of these tangential services has a tragic combination of being costly to pull off and, well, not a big idea.

While it saddens us that we are leaving so much behind in terms of what we built and developed for mail, we are equally excited to begin a new chapter in our company's life. Our team has been working on a new product that has already shown signs of success, and we believe it has the opportunity to be massively disruptive. We can't wait to tell you more

about it—but for now we're in stealth mode.

Our many learnings have led us to tackle a problem in many ways similar: a giant, sleepy industry that serves every American, is generally hated, and is in need of radical new solutions that involve hardware, software, and logistics.

Forefront in our minds are the learnings from the wild adventure of Outbox:

- Giant, complex systems appear insurmountable, but aren't—they were built by people just like you and me
- The main asset the government (and big companies) has is time—which is the resource of which startups have the least.
- You may think government organizations are completely, insanely backwards; you are wrong—they are worse.
- If you can't find a hardware solution to your needs, build it—it's not that hard.
- Doing extraordinary things for customers is time consuming and hard—but very worthwhile.
- Life is too short to pursue anything other than what you are most passionate about.

For the last two years, we got out of bed every morning because of the chance to re-imagine a daily activity for every American. We've had many sleepless nights these last few months, but are excited to be turning our complete attention to a new, equally compelling reason to wake up each morning.

With thanks, Will & Evan, Outbox Cofounders

Original article here: http://blog.outboxmail.com/post/74086768959/outbox-is-shutting-down-a-note-of-gratitude

Startup Failure: Exec

What I learned about online-to-offline

Many new online-to-offline entrepreneurs have asked me about my experience founding Exec. "Uber for X" businesses seem to be the startup of the day – it probably helps that on the face they are less technically complex and more accessible to aspiring less-technical founders.

Exec (subsequently changed to "Exec Errands") started off as us trying to fill my own personal desire for a part-time personal assistant / errand runner. I had enough random tasks that I wanted done and was willing to pay for, but didn't have enough to pay someone full time. I tried hiring a shared assistant, but it was hard to get her to run same day errands on late notice. I tried Taskrabbit and Craigslist, but the bidding and selection process was too high friction to use for smaller, every day tasks. Eventually, we came up with the first version of Exec, which we described as "Uber meets Taskrabbit", where anyone from the web or their iPhone could call an assistant who would immediately start running their errand.

On the backend, we had recruited a bunch of people looking for extra work to install our worker app and wait around for jobs. When someone put a job into the system, it would be offered to each errand runner in turn based on their proximity. When a job was accepted, the errand runner would get in touch with the customer, and start running the job. Early customers put in a wide variety of jobs, including dog walking, grocery pickups, temporary office work, and more.

Eventually, we dropped Exec Errands and turned Exec into a pure book-online house cleaning business, which was acquired by Handybook in January. Here are some of the lessons I learned in the early days of Exec Errands about online-to-offline businesses:

Unit economics matter a lot more than in pure software businesses. With Justin.tv, when we wanted to generate more revenue, we would look hard and figure out yet undiscovered places to shove another ad unit. Actual COGs (bandwidth and servers) were only ~30% of our gross revenue – most of our expenses were software engineers working on improving the product. With Exec Errands, we paid out 80% of the $25 / hour that we billed for errand runner time on job. That means that all marginal customer acquisition, customer service, and recruiting sufficient supply of errand runners had to be paid for out of that remaining 20%. Paying for mistakes (workers messing up a job) or customer refunds quickly ate up any profits.

Turnover of errand runners was very high. Most competent people are not looking for part-time work. The exception to this are people looking for supplemental income, but those people have full time jobs which conflicted with our high demand periods: mid-morning of the work week, Monday - Friday. Hiring new errand runners was expensive, and it was difficult to get them to stick around when we couldn't guarantee work – the average utilization (time on the job / time available) was only 50%. Additionally, most of our errand workers wanted to work during normal work days, meaning there was an undersupply on weekends.

Demand was very spiky. It turns out that $20 / hour does not provide enough economic incentive in San Francisco to dictate when our errand runners had to be available, leading to large supply gaps at times of spiky demand. Without some form of Uber-like surge pricing, it was impossible to ensure that we had consistent availability.

Customer acquisition was hard. Lyft and Uber are viral online-to-offline businesses, because many of the times you call a ride you take it with others (who subsequently think it's an incredible experience and download the app). Exec Errands was a not nearly as viral (and cleaning not at all), because you could use those services without ever telling anyone.

Customer activation was hard. Only about 10% of people who signed up for Exec Errands activated (put in at least one job request). When they heard about Exec, people thought it sounded like a cool idea, but rarely did they have an errand waiting in their back pocket that they hadn't yet run – making it hard for them to try us immediately. This led to reduced activation rates and high customer acquisition costs (since fewer signups turned into actual customers).

We shouldn't have run jobs ourselves. In the early days, if a job request came in and there were no Execs available to run them (as was often the case due to spiky demand), we sent someone from the office to run the jobs ourselves. We called that eating our own dog food and giving ourselves a sense of what could be improved about the product for both errand runners and customers. Unfortunately, this gave us a false sense that the quality of service for our customers was

better than it was, as the average competence of our employees was higher than our average recruitable errand runners.

Verticalize. Ironically, Exec was very similar to Justin.tv in this way: the product got a lot better when we started focusing on a specific vertical, and surprisingly, grew much faster. With Justin.tv, we discovered that a small number of users were very passionate about gaming, and thus we started working on Twitch, a community based around watching gaming video, which seemed like a very small niche to us at first. Now, Twitch has 45 million monthly viewers, and is many times the size of Justin.tv. I failed to learn the lesson the first time around, and started Exec Errands as a general errand running site that you could put any job into, from shopping to cleaning to getting your car washed. When you do many things, it is hard to do any one thing well, and Exec Errands turned out to be a merely adequate tool for many jobs, but not an exceptional one. When we started focusing only on cleaning, we were able to vet workers against a smaller skill set, provide specific tools, and streamline the interface to make purchasing more intuitive and simpler (and thus increase conversions), resulting in much better growth.

As a programmer entrepreneur whose previous businesses were purely online, those are some of the things I wished I had known when first starting work on Exec Errands. Hopefully this is useful for you.

Original article here: http://justinkan.com/exec-errands-post-mortem

Startup Failure: Stipple

Stipple Shuts Down

Image-tagging advertising startup Stipple Inc. has called it quits, shutting its doors less than 18 months after raising capital at a valuation just south of $25 million.

Backed by investors including Floodgate, Kleiner Perkins Caufield & Byers and entertainer Justin Timberlake, the company created tagging technology that allowed users to hover over digital images and dive into details provided by the brand, like cost or location.

"We had turned on revenue, but did not scale fast enough. We were not yet profitable," Stipple Founder and Chief Executive Ray Flemings said.

"Like many companies we got into the Series A crunch and we weren't able to raise more money," he added, referring to the explosion in early-stage funding several years ago and the follow-on investing that didn't keep pace.

Stipple announced it was ceasing operations on its Twitter account.

Founded in 2010, Stipple last raised $3 million as part of a Series A financing in 2012 to expand from Twitter to other social networks, according to VentureWire records.

At the time the company said it had tagged more than 80 million images and had more than 140 brand partners including Nordstrom, Nike, Zappos and L'Oreal. The plan in 2012 was to earn revenue through both display advertising and as a percentage of sales.

"We simply weren't able to get dollars flowing from the marketplace to line up with our expense structure," Mr. Flemings said, adding he is now exploring an asset sale. "It's never awesome, as an entrepreneur, to close your business."

Stipple had raised $10 million. Other investors included Sands Capital, Global Brain Corp., Parkview Ventures, Quest Partners, Thomvest Ventures and individual investors, according to VentureWire records. The company employed 14 and was based in San Francisco.

In December last year, Stipple competitor PICT was acquired by PopSugar. PICT had raised $1.3 million from Lowercase Capital, Forerunner Ventures, Opus Capital, AngelPad, 500 Startups and individual angel investors. PICT let advertisers and publishers turn images into "shop-able" content, meaning users can click on a product featured in a photo, for example, and buy it on the spot.

Other Stipple competitors include Luminate Inc. and ThingLink Inc.

Original article here: http://blogs.wsj.com/venturecapital/2014/04/22/stipple-shuts-down-vc-backers-included-floodgate-kleiner-perkins/

Startup Failure: Delight

Failed to delight - Post-mortem of my first startup

We decided to flip the switch on Delight at the end of this month and it officially marks the end of my first startup. Bittersweet moment.

The vision of Delight was to build a new kind of mobile analytics: visual analytics. Traditional metrics tell you the what and we tell you the why by showing you the real interaction. We had some good initial response but we failed to take off. Here are some lessons we learned along the way.

Price for the values your service delivers. Our most expensive monthly plan was US$300. Customers who churned never complained about the price. We just didn't deliver up to their expectation.

Customers pay for information, not raw data. Again, it's the value your service delivers. Customers are willing to pay a lot more for information and most are not interested in data. Your service should make your customers look intelligent in front of their stakeholders. It has always been on our roadmap to analyze the recordings so our users only need to watch the relevant ones. We never got to it though.

Price your service to encourage engagement. We originally priced by the number of recording credits. Since our customers had no control over the length of the recordings, most of them were very cautious on using up the credits. Plans based on the accumulated duration of recordings would have made much more sense for us and the number of subscription showed.

Follow up with inactive users. This is especially true when your service does not give intermediate values to your users. In our case, our users would need to install our iOS SDK, distribute the apps and wait for actual usage before they could gain any insights from their recordings. Our system should have been smarter about checking upon our users at various stages.

Welcome email works well. We got this idea from my YC batch mate Optimizely. Send an email to your new user and ask them about how they intend to use your service and where they heard about you. We got about 10% reply rate and gained insight on our target market and what pain points they wanted us to solve.

Content marketing works well. Most of our traffic was from the mentions in other reputable blogs. These blogs have a long lasting effect. We continue to have new users from a blog post written well over a year.

We didn't pivot hard enough. Although Delight was born out of our own pain point when developing our second product, it was still nerve wracking telling the team the ship had to change course mid way. The company was founded on changing mobile video consumption and we ended up betting the company on building developer tools. Even though we saw promising signs early on, we waited too long to fully commit to it. The team was constantly distracted and didn't know where the company was really heading.

I'm extremely proud of the team and what we've achieved together. Although at the end our technical issues have stopped Delight from being used in a production environment, the ultimate reason on why we failed wasn't the product nor the market.

It was me. I failed. I gave up.

It was my first startup and I was made a single founder a year after we started. I didn't know how to handle the roller coaster ride and it took a toll on not only my physical and mental health but also my loved ones. So, I called it quits.

Original article here: https://medium.com/@thomaspun/failed-to-delight-4de7ef031c3a

Startup Failure: Mochi Media

With 'No Meaningful Position Beyond Flash', Gaming Platform Mochi Media Will Close March 31

Mochi Media, a distribution and monetizing platform for Flash-based games that was acquired by Shanda Games for $80 million in 2010, is closing down, with all services ceasing March 31. The latest casualty in the decline of Flash, Mochi's shut-down is also a message to those businesses built on it to diversify or perish.

"If Mochi had a more meaningful position today beyond Flash, then there may have been a different path for the company going forward," CEO Josh Larson notes in a frank blog post published Friday.

If the news is a strong mark of the general viability of businesses that are focused too much on the shrinking world of Flash gaming, it doesn't look like it came as a surprise to Mochi Media specifically: one of the co-founders, Bob Ippolito, was among those who had been trying to prevent a closure of Mochi and had even tried to re-acquire the company from Shanda.

In a post on a Mochi Media user forum about the news, he writes:

"Nobody at Mochi wanted this to happen and there were parties interested in acquiring Mochi from them (including myself) for more than they'd make by dissolving it. They're simply not interested in making a rational decision here, and they certainly don't care about you all like we do (past and present Mochi employees). We've been trying to prevent this from happening for quite some time, but we failed to change their plans."

To be clear, Ippolito is no longer working at Mochi but appears to have remained engaged with company happenings from the sidelines.

In an email, Larson tells us that all Mochi employees (including himself) are now wrapping things up and looking for work elsewhere.

The news comes not just on the eve of GDC, but also at a time when China-based Shanda Games has been the subject of acquisition rumors, with one report claiming Alibaba might pay upwards of $3.2 billion for Shanda Games and its controlling shareholder Shanda Interactive. (To be clear, we have heard from a reliable source that this was unfounded speculation.)

As for Mochi Media, you might call it another casualty in the decline of Flash, in this case as a platform of choice among games developers that are today focused on iOS, Android, Steam and more.

"If Mochi had a more meaningful position today beyond Flash, then there may have been a different path for the company going forward," Larson notes in a frank blog post published Friday.

Indeed, while a lot of the Flash-based games, written for the web, seem to be designed to promote the equivalent gaming apps for iOS or Android (you can see one example pictured above), Mochi may have had little to no role to play in how those games were distributed across other ecosystems.

"Most 'Flash' game developers I knew from the halcyon days of Mochi services (2006-2010) have moved on to make games in HTML5, Unity, and Corona for platforms like Android, iOS and Steam," games developer Steve Fulton notes in a post on a Gamasutra community board.

Mochi Media had provided a few different services: for Flash games developers it offered a distribution platform on the Mochi site as well as through a network of other sites; for web publishers it offered gaming content to put on their properties; and for advertisers it provided a network for serving their ads within and around this content.

All of these are now winding down. Mochi says it has stopped accepting new games, and advertisers using self-serve accounts can no longer pay any more money in.

Flash these days seems to have diminishing importance as a platform for web development. By one estimate, today it's

used on only 14.7% of web sites, a decline of more than five percentage points compared to a year ago. (Unsurprisingly, Flash-maker Adobe published some figures about how vibrant Flash is in the gaming world — although these come from over a year ago.)

Mochi's origins speak to a richer time for Flash. Founded in 2005 by Jameson Hsu and Ippolito, Mochi Media was quick to see an opportunity in providing a platform to help independent developers distribute their content — a concept that companies like Apple and Google picked up and ran with in their app stores.

In Larson's words, the idea was "You focus on making a great game, and we'll take care of the rest." As he recalls it, at the time Mochi was founded, "Flash was a platform that held a lot of potential if developers could find ways to track, monetize and build better games. Together, Flash and Mochi provided an on-ramp to a career or business in game development."

It was a business model that found some traction with VCs, too, with Accel and Shasta both backing the company.

As Mochi Media grew, it also used its position to promote the Flash platform as much as itself, organising the Flash Gaming Summit and other events. (Notably, there was no Flash Gaming Summit in the last year.)

But ultimately, as the wider gaming industry matured, it went in a different direction, even for games or developers that may first have had their start with Mochi.

"We take great pride in currently seeing Ninja Kiwi's Bloons TD5, and Flipline Studios'Papa's Freezeria To Go among the Top Games Charts on iOS," Larson writes. "We love that at one time we shared a desk with Casual Collective which is now known as KIXEYE."

After the end of the month, online accounts, dashboards, and forums hosted by Mochi Media will no longer be available. If games distributed by Mochi are hosted on third-party sites, they will continue to work, but without Mochi services such as ads, Live Updates, Scores, Achievements and Analytics, the company says.

Customers can download all of their data, and Mochi says it will also be sending basic data around views and revenue information for games to users before the end of this month. It also has made some details about how it plans to pay out remaining monies to customers for those with over $100 in their accounts.

Mochi has said that it has thousands of developers on its books. "Please keep in mind that a large number of people will likely be requesting a massive amount of data from our servers over the next couple of weeks, so please be patient with the computing times," Larson writes.

Original article here: http://techcrunch.com/2014/03/16/with-no-meaningful-position-beyond-flash-gaming-platform-mochi-media-will-close-march-31/

Startup Failure: HowDo

The last step

In 2012 we believed there was a fun, more human way to share the knowledge of how we make things. Excited to bring a voice to inspiring people, we set out to make it possible for anyone to share their experience of making; a community to shape the future of making and learning together.

That belief was not wrong, you have been the most brilliant, creative community. Our goal was now to transform that passion into a sustainable platform. We have failed to make this possible and without the resources needed for development, today How.Do comes to its last step.

Together we created more than 1.7 million learnings, and over 7 thousand projects. Discovering teenagers who made us laugh, professionals levelling-up our craft skills or the occasional whirlwind of electronic prowess - it's been a delight waking up everyday to new creations from all over the world.

You brought the platform to life with your self expression, and it is to you we are deeply sorry. We're wonderfully proud of the product and culture we built together, but it is your creativity that will be missed most.

As a team we will move onto new challenges. Thank you for letting us build this platform with you. Continue inspiring our world to be a more creative, fun and unique place.

Here's to the future. The How.Do Team\

Original article here: https://www.how.do/bye

Startup Failure: Readmill

Epilogue

Three years ago, we set out to create an open, independent reading platform. We believed that the reading experience could be beautiful and that it was meant to be shared.

Along the way, we created a reading app for iOS and Android that proved these things to be true. We're proud of the product we built, but even more so, we're grateful for the community of readers that made it grow.

At every turn, your feedback shaped Readmill's development, and your passion signaled a new chapter for reading. Together we wrote in the margins of ebooks and discussed our favorite passages from across the world. Thank you for helping to bring this reality into view.

Readmill's story ends here. Many challenges in the world of ebooks remain unsolved, and we failed to create a sustainable platform for reading. For this, we're deeply sorry. We considered every option before making the difficult decision to end the product that brought us together.

As of today, it is no longer possible to create a new account, and on July 1, 2014, the Readmill app will no longer be available. For the next three months, our first priority is to help you transition to other services and get back to reading. All of your books and reading data are available for export in multiple formats.

Our team will be joining Dropbox, where our expertise in reading, collaboration and syncing across devices finds a fitting home. Millions of people use Dropbox to store and share their digital lives, and we believe it's a strong foundation on which to build the future of reading. We're delighted to work alongside this talented team and imagine new ways to read together.

Thank you for the time and attention you've contributed to this community. It has been a privilege to read together, and we look forward to meeting again, in new ways, in the margins.

Sincerely, Henrik, David & the Readmill team

Original article here: https://readmill.com/

Startup Failure: Plancast

The Uphill Battle Of Social Event Sharing: A Post-Mortem for Plancast

Editor's note: Mark Hendrickson is the founder and CEO of Plancast, a social site for planning events, which he has decided to stop working on full-time. In this guest post, Hendrickson takes us through a detailed analysis of why it never took off and what he learned. He is also a former TechCrunch writer.

Nearly three years ago, I left my position at TechCrunch to start my own Internet business, with the idea of creating a web application that'd help people get together in real-life rather than simply helping them connect online as most social networking applications had done.

Plancast was the service conceived a few months later from that basic inclination. Its approach was to provide a really easy way for people to take whatever interesting plans they had in their calendars and share them openly with friends, with the rationale that greater social transparency for this particular type of personal information would facilitate serendipitous get-togethers and enable a greater awareness of relevant events. Personally, I figured that knowing more about the events my friends and peers were attending would lead to a more fulfilling social and professional life because I could join them or at least learn about how they spent their time around town.

Along the way my team built a minimum viable product, launched from obscurity here on TechCrunch, raised a seed round of funding from local venture capitalists and angel investors, and worked like mad to translate our initial success into long-term growth, engagement and monetization.

Alas, our efforts began to stall after several months post-launch, and we were never able to scale beyond a small early adopter community and into critical, mainstream usage. While the initial launch and traction proved extremely exciting, it misled us into believing there was a larger market ready to adopt our product. Over the subsequent year and a half, we struggled to refine the product's purpose and bolster its central value proposition with better functionality and design, but we were ultimately unable to make it work (with user registration growth and engagement being our two main high-level metrics).

This post-mortem is an attempt to describe the fundamental flaws in our product model and, in particular, the difficulties presented by events as a content type. It's my hope that other product designers can learn a thing or two from our experience, especially if they are designing services that rely on user-generated content. The challenges I describe here apply directly to events, but they can be used collectively as a case study to advance one's thinking about other content types as well, since all types demand serious analysis along these lines should one seek to design a network that facilitates their exchange.

Sharing Frequency

Social networks (by my general definition and among which I count Plancast) are essentially systems for distributing content among people who care about each other, and the frequency at which its users can share that content on a particular network is critical to how much value it'll provide them on an ongoing basis.

Unlike other, more frequent content types such as status updates and photos (which can be shared numerous times per day), plans are suitable for only occasional sharing. Most people simply don't go to that many events, and of those they do attend, many are not anticipated with a high degree of certainty. As a result, users don't tend to develop a strong daily or weekly habit of contributing content. And the content that does accrue through spontaneous submissions and aggregation from other services is too small to provide most users with a repeatedly compelling experience discovering events.

I run the service, and even I currently have only five upcoming plans listed on my profile, with a total of 500 plans shared over the last couple of years, in contrast to almost 2,800 tweets on Twitter over the same period of time. People often tell me "I like Plancast, but I never have any plans to share". With social networks, this is sometimes a case of self-awareness (such as when people say they don't know what to tweet), but often they're simply telling the truth; many Plancast users don't have any interesting plans on their calendars.

Consumption Frequency

People also don't proactively seek out events to attend as you might suppose. I've gotten into the habit of thinking about people as divided into two camps: those who have lots of free time and those who don't.

Those who do are often proactive about filling it, in part by seeking out interesting events to attend in advance. They are generally more inquisitive about social opportunities, and they will take concrete steps to discover new opportunities and evaluate them.

Those who don't have much free time often desire to conserve it, so rather than seeking out or welcoming additional opportunities, they view them as mentally taxing impositions on a limited resource. For them, planning is a higher-risk endeavor, and usually they'd rather not plan anything at all, since if they're busy, they likely have a preference to keep their free time just that – free.

It's hard to generalize by saying most people are in one camp or the other, but suffice to say, there are many people in the latter. And for them, it's hard to get them excited about a service that will give them more options on how to use their time.

Tendency to Procrastinate

Even putting this bifurcation aside, most people resist making advanced commitments before they absolutely need to make them. People fear missing out on worthwhile events but don't actually like to take the deliberate initiative to avoid such missed chances, which requires planning.

This can be attributed primarily to people's desire to keep their options open in case other conflicting opportunities emerge as the date and time of an event approaches. If they can afford to wait and see, they will. Therefore, their commitment will be secured and shared in advance only when they're particularly confident they'll attend an event, if they need to reserve a spot before it fills up, or if there's some other similar prerogative.

Incentives to Share

Returning to the topic of sharing plans, it's not only a matter of having interesting plans to share but being compelled to actually share them. And unfortunately, people don't submit information to social networks because they love data set integrity or altruistically believe in giving as much as possible. They do it because the act of contribution selfishly results in something for them in return.

Most social networks feed primarily on vanity, in that they allow people to share and tailor online content that makes them look good. They can help people communicate to others that they've attended impressive schools, built amazing careers, attended cool parties, dated attractive people, thought deep thoughts, or reared cute kids. The top-level goal for most people is to convince others they are the individuals they want to be, whether that includes being happy, attractive, smart, fun or anything else.

This vanity compels folks to share content about themselves (or things they've encountered) most strongly when there's an audience ready and able to generate validating feedback. When you post a clever photo on Instagram, you're telling the world "I'm creative!" and sharing evidence to boot. Those who follow you validate that expression by liking the photo and commenting positively about it. The psychological rush of first posting the photo and then receiving positive feedback drives you to post more photos in the hope of subsequent highs.

Sharing plans, unfortunately, doesn't present the same opportunity to show off and incur the same subsequent happy feelings. Some plans are suitable for widespread consumption and can make a person look good, such as attending an awesome concert or savvy conference. But, frustratingly, the vainest events are exclusive and not appropriate for sharing with others, especially in detail.

The feedback mechanisms aren't nearly as potent either, since coming up with a worthy comment for an event is harder than commenting on a photo, and "liking" a plan is confusing when there's also an option to join. The positive feedback of having friends join is itself unlikely since those friends have considerations to make before they can commit, and they'll tend to defer that commitment for practical purposes, per above.

Additionally, if a user wants to show off the fact they're at a cool event, there is little additional benefit to doing so before the event rather than simply tweeting or posting photos about it while at the event. An important exception is to be made for professionals who style themselves as influencers and want to be instrumental parts of how their peers discover events. This exception has indeed been responsible for much of our attendee-contributed event data among an early-adopter community of technology professionals.

Selectivity & Privacy Concerns

Vanity, of course, is not the only possible incentive for users to share their plans. There's also utility to getting others to join you for an event you'll be attending, but this turns out to be a weak incentive for broadcasting since most people prefer to be rather picky about who they solicit to join them for real-life encounters.

While event promoters have a financial interest in attracting attendees far and wide, the attendees themselves mainly turn to their closer circle of friends and reach out to them individually. You don't see a lot of longer-tail plans in particular (such as nights out on the town and trips) because people are both wary of party crashers and usually uninterested in sourcing participants from a wide network.

The Importance of an Invitation

On the flip-side of this reluctance to share plans far and wide is the psychological need for people to get personally invited to events.

Plancast and other social event sharing applications are rooted in an idealistic notion that people would feel confident inviting themselves to their friends' events if only they knew about them. But the informational need here is not only one of event details (such as what's going to happen, when, where and with whom). People often also need to know through a personal invitation that at least one friend wants them to join.

When you have a service that helps spread personal event information but doesn't concurrently satisfy that need, you have a situation where many people feel awkwardly aware of events to which they don't feel welcome. As a result, the most engaging events on Plancast are those that are open in principle and don't solicit attendees primarily through invitations, such as conferences and concerts, where the attendance of one's friends and peers is a much less important consideration for their own.

Content Lifespan

Getting content into a social network is not enough to ensure its adequate value; there's also an importance of preserving that content's value over time, especially if it just trickles in.

Unfortunately, plans don't have a long shelf life. Before an event transpires, a user's plan for it provides social value by notifying others of the opportunity. But afterwards, its value to the network drops precipitously to virtually nothing. And since most users don't have enough confidence to share most plans more than one or two weeks in advance, plans are typically rendered useless after that length of time.

Contrast this expiration tendency with more "evergreen" content types, such as profiles and photos. Other people can get value out of your Facebook profile for years after you set it up, and the photos you posted in college appear to have even increased in value. Nostalgia doesn't even have to play a part; people's hearts will melt upon viewing this puppy on Pinterest, Tumblr, and other visually-heavy content networks for a long time to come. But how much do you care that I attended a tech meetup in New York last October, even if you're my friend?

Geographic Limitations

Geographic specificity is another inherent limitation to a plan's value. Unlike virtually all other content types (with the exception of check-ins), plans provide most of their value to others when those users live or can travel near enough to join.

I may share plans for a ton of great events in San Francisco, but few to none of my friends who live outside of the Bay Area are going to care. In fact, they'll find it annoying to witness something they'll miss out on. Sure, they might appreciate

simply knowing what I'm up to, but the value to that kind of surveillance is rather modest all by itself.

This is especially problematic when trying to expand the service into new locations. New users will have a hard time finding enough local friends who are either on the service and sharing their plans already, or those who are willing to join them on a new service upon invitation. People who encounter the service from non-urban locations have the hardest time, since there aren't many events going on in their area in general, let alone posted to Plancast. Trying to view all events simply listed within their location or categories of interest yields little for them to enjoy.

Looking Forward

Despite all of these challenges, I still believe someone will eventually figure out how to make and market a viable service that fulfills our aims, namely to help people share and discover events more socially. There's simply too much unearthed value to knowing about much of what our friends plan to do to leave information about it so restricted to personal calendars and individuals' heads.

Another startup may come along that develops insight into an angle of attack we missed. Or, perhaps more likely, an established company with an existing event or calendaring product will progressively provide users with a greater ability to share their personal information contained within. On the calendaring side, Google is possibly the best-situated with Google Calendar and Google+, which together could make for a very seamless event sharing experience (one of the things we considered seriously for Plancast was deep personal calendar integration, but a sufficient platform for it simply wasn't available). On the events side, companies like Eventbrite, Meetup and Facebook have services that are primarily compelling for event organizers but already contain useful data sets that could be leveraged to create their own social event discovery and sharing experiences for attendees.

Plancast managed to attract a niche audience of early adopters who found it to be among the most efficient ways to share and hear about events (thanks, users! you know who you are). Over 100,000 have registered and over 230,000 people visit each month, not to mention enjoy the event digests we send out by email each day. For that reason alone, and despite its growth challenges, we're going to keep it up and running for as long as possible and are hopeful we'll find it a home that can turn it into something bigger. It's my expectation that one day mainstream society will take for granted the type of interpersonal sharing it currently enables for just this small community, and I look forward to seeing how technological advancements overcome the aforementioned challenges to get us there.

Original article here: http://techcrunch.com/2012/01/22/post-mortem-for-plancast/

Failed Startup Lessons

7 Lessons I've Learned from a Failed Startup

This week I had the VERY fortunate opportunity to take part in Propel ICT's Launch36 startup bootcamp. The speaker of the event was Ash Maurya (@ashmaurya), founder of Spark59 and author of the lean startup book called Running Lean. He discussed how to apply lean methodology in startups using his Lean Canvas.

I attended this bootcamp as a startup applicant for the launch36 Accelerator program with Propel ICT (@propelict). And I'll be pitching my new startup called Recommendur (click here to signup for exclusive beta access) to a group of angel investors on February 4th.

If you didn't know, I've already been the co-founder of one FAILED startup. And yes, I am extremely proud of it.

It pushed me so much further than I could have ever imagined. The lessons that I learned were invaluable and priceless. If you don't mind, I'd like to share my startup experience with you. My Startup Experience

First off, I had THREE jobs.

Yes, three.

Why?

Well at that time, I had only begun to consult in a professional capacity. And as many of you know, it takes time to get your name out, so I needed to supplement my income with a secondary job. A job as night auditor at a local hotel to be exact. Quite a switch from working for a Fortune 500 company.

Here's what an average day looked like:

7am: Arrive home and go to sleep. 12pm: Wake up and lunch 1pm to 9pm: Startup work/ Consulting 9pm to 10pm: Power nap 10pm to 11pm: Get ready and drive to work 11pm to 7am: Night Audit

Then it started all over again. I did this for 7 months straight. And it nearly killed me. BUT….I don't regret a minute of it. See, a failure is worth SO much more than a success. When you fail, it forces you to ask the hard questions. Why did it fail? Did I not try hard enough? Did I not do my homework? Wrong timing? You have to re-evaluate yourself, your priorities, your reasons, and your motivations. And the best of all, it teaches you what NOT to do in your next one! ;)

7 Lessons Learned From a Failed Startup Here are 7 out of many lessons that I've learned though my past startup experience:

Never Assume, Always Validate

This is so important. You need to validate that your product or service solves a real problem, and whether your target market is willing to pay for your solution. In other words, test your idea before dumping a huge amount of cash on it.

When we launched our startup, we did neither. Because of the market we were targeting, it was difficult to validate (at least we thought so). So we launched with the hopes of validating as we went along. Bad idea.

A Good Co-founder is Worth Their Weight in GOLD:

Having the right co-founder is extremely important. My co-founder was an excellent complement and the voice of reason. There were days where I wanted to throw in the towel, and he helped me stick with it. As well, there is a bond that happens. You can be brutally honest with each other and they won't judge you. At least that was my experience.

Always Have a Technical Co-Founder/Equity Partner:

Trust me, it is SO much harder to get things done when you don't have someone in your team capable of doing the

technical work (software development). Fortunately we managed to secure one great developer for awhile, but not having a developer as part of the team really held us back.

You Are MUCH More Resilient Than You Think

Had someone told me I was going to work a night audit job, get 4 hours sleep a day, consult, and be part of a startup, I would have told them that they're crazy. I knew I was resilient, but this really pushed me to the next level. NEVER underestimate your capabilities.

Never Be Afraid To Call Yourself a Startup

For some reason when we launched, we didn't want to be labeled as a startup. Initially our reasoning was if we called ourselves a startup, we'd be perceived as less stable or reputable. What we failed to see was the enormous amount of support that was available for startups.

Never Be Afraid To Ask For Help

When you're in a situation or scenario, don't be afraid to ask others for help. Just make sure that those you turn to have "earned" experience and have lived through a similar scenario. In other words, don't simply turn to your family and friends for advice. Most often you get the watered down version, simply because they love and support you, and most likely don't want to hurt your feelings.

Oh, and often times when it feels like everything is falling apart, it's not. :)

7."Brick and Mortar" Products Can Sink Your Startup

When we launched, we bought products that we "thought" our target market would purchase. And it was quite expensive. NEVER buy physical products to resell unless you've validated that your market will pay for it, and it is profitable.

Original article here: http://www.brightcube.ca/blog/7-lessons-ive-learned-from-a-failed-startup/

Startup Failure: Intellibank

7 Things I Learned from Startup Failure

My former company, Intellibank, was sort of like Dropbox done wrong. You've probably never heard of Intellibank, because it came and went like so many startups do, but it was a promising company with smart people. We raised money and could have been the next big thing, but it never happened. Why?

Though Intellibank was not successful, I don't view my time there as wasted. From the mistakes we made, I learned what not to do — and from there, I've arrived at some essential truths that can lead to startup success.

It's all about the market

Even the best team with the best product will fail if its market does not exist. Venture capitalist Marc Andreessen says that product-market fit (a term he is said to have coined) is one of the most important factors he considers when evaluating startups.

At Intellibank, we did not achieve product-market fit. Every customer was asking for something different and we gave it to them. We had six markets with 40 different types of customers, and in hindsight, we should have developed just one product. We couldn't be all things to all people — and by failing to declare our major, we created a world of chaos for our sales, product and marketing teams.

Even if you do have product-market fit, you will not get very far if the market is not big enough. To determine whether you fall into that category, ask yourself if the market you're targeting is big enough to allow for pervasive adoption of your product and exponential growth.

Validate with your customers, not your investors

A common mistake among entrepreneurs is seeking validation of their ideas and decisions from investors. The most important people any company should seek validation from are their customers. That's right, your customers matter more than your investors — and any good investor would agree.

Do not ignore yellow lights coming from your early adopters, because their activity indicates a momentum shift. Spend time understanding all aspects of the customer value proposition. I once heard Guy Kawasaki talk about his 10X rule — in order for people to switch and buy your new thing, your product doesn't need to be perfect, it just needs to be 10X better than the alternative. Think about the 10X rule and ask yourself: why should your customer buy your product? How does your product fit into the rest of his world? What influences their opinion of the product's value? What is your product displacing — all products displace something — and why should your customer risk making that switch? You need to be as knowledgeable about your customer and their needs as you are conversant with your own product.

Focus, focus, focus

Focus and simplicity are often more difficult to achieve than building features on top of features on top of features. As a result, too many startups are unfocused. The time required to trim back an idea is not insignificant — said best by Mark Twain: "If I had more time, I would have written a shorter letter."

In order to succeed, a startup needs to do one or two things exceptionally well; some of the greatest products today don't have a million bells and whistles, but they solve one concrete problem brilliantly. I'm thinking of companies like Salesforce.com in the early days. They entered the market with good products, and over time they iterated, grew and added features — but not before they owned essentially the entire market.

By keeping it simple, measurable and achievable you'll be well on your way. Everyone at your company should be able to articulate the goal of your business, enabling a dogged, unyielding focus on that goal throughout the organization.

Aim to exceed expectations

Your goal should not be meeting your customers' expectations; it should be exceeding them. Truly great and memorable products surprise and delight their customers, so don't be afraid to spend the time and money to build an exceptional product. But don't let this pursuit inflate your product's ego, if you will — making promises you cannot keep will leave you surrounded by disappointed customers, investors and employees.

I cannot emphasize how important it is in the long run to over-deliver to your customers. For example, Fab does not have to give every customer a $5 gift card with every order, but doing so wins them a lot of brand loyalty and even word-of-mouth marketing.

Figure out streamlined metrics to measure your progress

I once had a board member tell me that we were over-measured and under-prioritized. It stung. A lot. But it also made quite an impression. As a business leader you need to figure out the metric that matters most for your company and understand that the more you measure, the less prioritized you'll be. Don't fall into the trap of trying to measure everything. What I've learned is that in the early days, what matters most is having customers who love and use your product. Figure out the one or two best measures to determine this.

Pivoting is okay… but it is not a business strategy

I learned this one the hard way. At Intellibank, we would change our pitch deck based on what we thought would get us traction with investors. In one particular meeting, I was in the middle of explaining our revenue model, when a potential investor interrupted me and asked, "Can you tell me what your product actually does?" We were pivoting so often for different types of customers that we completely lost the big picture. You must be agile, but not to the point of an identity crisis; you have to look beyond your four walls and convey the big picture.

Ultimately, deliver a great experience. It's what keeps people coming back

Customers come to restaurants for a great overall dining experience, but the food is the baseline. They come back if the service and experience exceed their expectations. It's the same with any business—the product is table stakes and it's the experience that brings people back.

Take a look at Hotel Tonight. It's only accessible from a mobile phone and while there have been several times mid-flight when I've wanted to use the app to book my hotel in my destination city, I can't and so I wait until landing. Using their service on my laptop would be nice, but the experience is so elegant and over-delivers every time, so I prefer to book my hotels there.

With all of these learnings in mind, think about the product you're selling and think about where you see it going. Now take a step back and ask yourself the most important question of all—when your customers are using the product how do they feel, and will that feeling keep them coming back?

Original article here: https://www.linkedin.com/pulse/article/20130923123247-758147-7-things-i-learned-from-startup-failure

Startup Failure: Teamometer

Here are the startup lessons learned from Teamometer, a startup with 2 years and about 300 users.

Two years ago, on December 2011, I was generating ideas for a business that would help team managers to not suck so much at managing their teams. I came up with this idea because I had some pretty terrible managers in my life. At the same time I worked for about 5 years with leadership development and had some pretty great teams and team experiences. Exactly January 1st 2012 I registered the domain teamometer.com.

Validating the MVP

I was reading The Lean Startup at that time, which is a great book, but not completely fool proof (here is a fool who got it wrong). I started well: in December I created a very basic idea and rough screens of my team management web tool. In January I built a very basic 2 page website with a video explaining what Teamometer.com did and a "try it free" button, which then lead to a page for inputting the email address to capture leads (since I had no product, that was just a test to see if there was demand). Then I bought a few dollars in Adwords to drive traffic to it.

What was good: create a simple and fast website was sweet. The presentation video was pretty good too in my opinion. The Adwords was good to drive traffic too and got a fair share of people who gave me their email address (about 10% of people who clicked the ad).

What could be even better: "try it free" and free trials in general are good if you know your product rocks and you will convert the customer. However, when you are testing demand, it is not the best, because it invites people that are only curious, not really thinking of buying. So the best approach would be to put a price page, see who would click that and after capturing the email telling that the product is not ready and they won't be charged anything just yet. I would probably get fewer prospects, but the people who did subscribe would be serious about the product.

Where it went wrong: I had all those people who were interested in the product. I judged the product "validated" (WRONG! WRONG! WRONG!) and went on to build it. Looking back, I cannot believe how stupid I was. I had all those people who said they were interested and the obvious next step was NOT building the product, but TALKING to the people who were interested. Asking them things like:

What were you looking for in Google when you found Teamometer? Why were you looking for it? Why is it a problem for you/company? How are you trying to solve this problem today? After discovering their needs, going through with them on what is your idea, to get feedback if you are on the right track.

If I did this, I would have a much better perspective of what people wanted.

Where it went REALLY wrong: one of the prospects was an HR person from a huge Indian manufacturer. They wanted the system NOW and wanted to speak to me. I had a Skype call with them and what they wanted was clearly not what I was thinking to provide with Teamometer. Instead of surfing the wave and adapting my idea to what a real prospect client was telling me they wanted, I entered "sales mode" and sold them my idea, what it would be, how good it was. How dumb is that? I just needed to build what they wanted.

The worst part is that I was blind, thinking that this call had validated my product even more, when in fact it was clearly invalidating it.

Finding partners to build the idea

Once I believed the idea was validated (it was not), I started to scramble to find a way to make the product. I spoke with tons of developers, but my persuasion skills were not impressive enough. I got none. Then an old friend, which founded a million startups, told me he was interested and he knew a guy who was a developer.

Things moved pretty fast, and in no time we were 4 partners: me, the developer, my old friend (background in finance and marketing) and the brother of the developer (background in marketing). And when I say pretty fast, the whole thing

literally happened over a cup of coffee, the first time I met the 2 brothers. I needed a developer, I ended up with a developer and 2 business guys. BUT the 2 business guys were also responsible for financing the project.

There are so many lessons to be learned from this and so many stupid mistakes.

Founder roles and expectations. Conflict and disagreement arrived early, because expectations of work and founder roles were not agreed upon and everyone had a different idea about their share of the load:

The developer had no intention of being the project's developer (?) he was not really a developer, he was a computer science graduate who owned a webdev shop and was used to managing, not coding.

Everyone but me had other businesses, so time was constantly an issue and deliverables just didn't happen timely (which is pretty essential for a fast moving startup).

Since we were 3 business people, we spent all this time into idiot plans, budget forecasts, BUSINESS CARDS, fancy website… all useless things which in the end did not contribute to anything.

The result of this was that in the end we had to hire a full-time (and paid) developer. So we had zero revenue, 4 co-founders and a paid employee (which was effectively the only one doing real work).

Must I tell you that this was a disaster?

The first partner left the company a few months in. The 2nd one year after. And finally we ended our run after 2 years of zero paying customers.

Learning: I believe a company is the closest to marriage with kids among parents who (probably) don't have sex or have any sort of romantic love for each other. If marriage with sex and love is already tough, imagine now without any of those things. I should have been much more careful with who to bring in the company. And, if I was convinced of their value, I should have agreed really really well what was expected from each partner.

Never underestimate expectations.

By this time what is really import to hammer through your head is that building the product at this point was a mistake (it was not validated, we needed to speak to prospects). But I made more mistakes which are meaningful:

Finding a painful problem

Literally 100% of the feedback I got from free trial users was that Teamometer was useful and it helped them discover the issues in their team and tackle them. That was exactly what the tool should do, great! So what is the explanation for exactly 0% conversion rate from free trial to paid customers? Why were we not able to sell it?

Simply put, because we didn't have a painful enough problem.

If a company stops invoicing, they will immediately feel the pain. If a company stops managing well their teams, it can take as much as 1 year for people to realize there is something wrong – and they won't know what is wrong. Maybe some people will leave, but it is easy to put the blame on the employee or the market.

It is possible to sell soda in the desert (nice to have), but it is much easier to sell water (must have).

(Don't) multiply big numbers

Multiply $30 times 1.000 clients times 24 months. WOW, we will be rich!

Oh, silly you, you have no idea how hard it is to get 1.000 clients paying anything monthly for 24 months. Here is my advice: get your first client. Then get your first 10. Then get more and more.

Until you have your first 10 clients, you have proved nothing, only that you can multiply numbers.

Forget business cards

…and all the crap about registering company, etc etc etc. While you have no sales (product validation), you should not invest in such gimmicks that only distract from the main goal, which is to find a repeatable business model.

SEO and social media bullshit

Related to the above: I wrote a damn article every effing day. It made us jump to the first page of Google in several important keywords. How did that translate to sales? Zero.

So the lesson is (unless your product is a multi-sided business like Facebook, where users are not paying customers) do not invest time and money to get more traffic. If you do, make sure to TALK to those people, because validating the product is more important than vanity metrics like how many likes you got on Facebook.

If there is just one thing you should learn, it is: Just speak to prospects and extract their pain, then sell the painkiller (before building the product). If they are willing to buy, do take their money and invest that money into building the product.

Original article here: http://www.sergioschuler.com/startup-lessons-learned-from-my-failed-startup/?hvid=3EvjLn

Startup Failure: Standout Jobs

A Postmortem Analysis of Standout Jobs

It's been a number of months since I announced that we sold Standout Jobs. In that time I've gone through a myriad of emotions and cyclical arguments in my head about the entire process of starting Standout Jobs, running it and selling it. I've read a number of great (and heartfelt) postmortems of late, and then read Eric Ries' blog post: Stop lying on stage. I appreciate Eric's point-of-view, and I've had similar experiences watching people on stage, publicly addressing an audience and omitting key points of their "startup's story". Investors do it too. Everyone does it (I think it's human nature.) But there's a big difference between "on the record" and "off the record". People should be smart enough to know the difference. And there are some bridges not worth burning (at least completely.) Even more importantly, there's just some things that aren't meant for public consumption. Perhaps in those cases the stories (or "legends!") shouldn't be promoted whatsoever, because they're only half-truths...

One thing is certain: Success is never as easy as it seems when someone else talks about it, and failure is rarely as dismal as it seems either.1

"Overnight successes" are generally bullshit. No road to startup success is without big, massive road bumps. It's called a rollercoaster for a reason. On the flip side, failure is rarely so crippling a blow that entrepreneurs can't pick up and keep going. Everyone fails. "Out of the ashes..." and all that.

So in reflecting on all of this (and more), I've decided to write the Standout Jobs postmortem. And the only way to do so effectively, publicly and hopefully with the end result of helping other entrepreneurs, is to reflect on my own performance and efforts.

Although we did exit Standout Jobs it was not a financial success. In fact, personally, I have less money now than when I started the company. Initially (I can't remember for how long) I wasn't paying myself. When we raised financing, I paid myself a fair salary, but then cut it down when the company was struggling. At the end, I wasn't paying myself for quite awhile as well. Such is life when you decide to start a company. High risk = high reward (or not!)1

I am pleased (as I indicated when I announced the exit) that we found a buyer for Standout Jobs. It was a great and difficult experience to go through. It was a true learning experience. The technology we built was solid, and I do honestly hope that the acquirer is able to leverage it with great success. You can certainly see in the marketplace today that a number of players have been working on enhancing employer profiles, increasing employer branding and social recruiting opportunities for employers. We were onto something.

Timing Sucked

Our timing was terrible. We launched the paying version of our application in the fall of 2008 about 5 minutes before the economy collapsed. Very few companies were hiring. I got feedback on sales calls like this: "That's a great product, really love it, but we won't be hiring for another 18 months or so. You have anything to help us fire people?"

We knew the economy was heading south, but we had no idea how bad it would be and how long it would last. And we failed to react quickly and aggressively enough. In fact, I think most failures of CEOs and startup founders fall under the category of, "didn't react quickly and aggressively enough." I'd say it's fairly rare that a startup overreacts to something and changes too radically. In 99.9% of cases it's the opposite: startups don't change fast enough. But making aggressive, fast changes takes bold, bold leaders.

Not Bold Enough

Simply put, I wasn't bold enough. It took too long for me to sound the alarms loud enough. There are a number of reasons for this, and I'd say they're quite common. For starters, I'm not a loudmouthed egomaniac. I don't think you have to be, but you do need a big ego to survive. I have a healthy enough ego (most days!) but it did take a beating during the course of Standout Jobs. It's not that hard to get to a point where you believe that everyone else knows better than you do. And having to tell your investors, Board of Directors, employees, family, etc. that what you told them would work isn't working is just plain hard.

I absolutely believe that taking a more analytical approach – driven by Customer Development and Lean Startup principles – can bolster your sagging ego and help you make rational, honest and difficult decisions. With Standout Jobs, I wasn't rigorous enough in this process. I won't make that mistake again.

Market Understanding

I didn't have a strong enough understanding of the HR / Recruitment market going in. I've written about this before and I see countless entrepreneurs make the same mistake. They look at a market objectively and think, "I can fix that!" only to realize when they get neck-deep into it that there are a whole bunch of issues they didn't understand.

I just didn't do enough homework. And doing homework while you're writing the test as fast as you can is pretty damn hard.

Having said that, I do think that I did a good job pushing my way into the industry leadership, networking, communicating Standout's vision and catching a good sense of where the market was heading. And in some markets, brand awareness and brand building amongst the evangelists and leaders is critical to gaining exposure and generating buzz. It can provide a false sense of success, which is something to be wary about, but it can also add significant value.

The lack of market understanding ultimately meant that we couldn't match the right product to the right market at the right price.

Product Development

I had an exceptionally strong team. Best I've ever worked with. But in reality we didn't code and launch fast enough. We didn't get product into customers' hands fast enough. We were fast, but next time I'll move faster, and iterate much more frequently. Of course that's easy to say, but I think a lot of developers aren't as comfortable as they need to be with this strategy. The fact is that having a specification and building to that specification is a lot easier for a developer; constantly changing requirements, moving things around, cutting features, adding stuff, circling back with feedback, etc. makes a developer's job harder. More rewarding. But harder.

In the future I'll focus a lot more on a Minimum Viable Product (MVP) and getting it into the hands of customers as early as possible. And even more importantly, I'll solicit real, hard feedback from them. With Standout Jobs, we did things in a slightly incorrect order, which distorted our view of reality.

Shoving Things Down the Market's Throat

If you find yourself in an argument with a market, you're very likely to lose. As we got Standout Jobs into the hands of customers we saw a few troubling issues. People weren't using it as actively as we had expected. People were still following the same bad habits they had before even though we provided them a "better way." Ultimately changing behavior is extremely hard. It's possible, but you need to be very, very creative about it.

I didn't realize just how much of a behavior change we were asking people to make. We did find ways of circumventing people's unwillingness to change, but we never committed to those strategies fully, in part because they didn't work quickly enough, and in part because we were stubborn.

Simplifying the Value Proposition

I was able to simplify Standout's value proposition to the point where it was clear, but that's not always enough. Part of the problem was that our value proposition lacked immediacy: "Invest in your employer brand and over time you'll hire better fits into your company at a faster rate." The problem is with "invest in your employer brand" – too few people truly have time for that (although they should!)

I didn't make a strong enough case for tying the value proposition to something immediate that HR could sell internally to their bosses. It's important to note that "selling a value proposition" and "delivering actual value" are two different things. I remember a sales call once with two people in an HR department. This was the second call. We were doing a web demo. As I was presenting, the new person on the call said, "We already have something like this. It's [name not included here]. It does what you're talking about." I was surprised that the first person I had met didn't bring this up since the other

product was somewhat competitive. It turns out that neither of the two people even knew the URL of the site that the competitor had launched for them. The competitor had managed to sell their solution to this company, and the company didn't even know where that solution was online. That was surprising and demoralizing. The competitor had managed to "sell their value proposition" extremely well, but whether they were delivering on value was a different story. Please note: I'm not arguing that you should sell a good value proposition and not deliver actual value, but you can't get to delivering actual value if you can't sell a good value proposition.

Value propositions need to be super clear, straightforward and deliver (or at least claim to deliver!) immediate results. Tying value propositions to making more money is best. Saving money is second best. Everything else is tertiary…

Raised Too Much Money, Too Early

I raised too much money, too early for Standout Jobs (~$1.8M). We didn't have the validation needed to justify raising the money we did. Part of the reason for this is that the founding team couldn't build an MVP on its own. That was a mistake. If the founding team can't put out product on its own (or with a small amount of external help from freelancers) they shouldn't be founding a startup. We could have brought on additional co-founders, who would have been compensated primarily with equity versus cash, but we didn't.

I had never raised venture capital before. Raising the money felt like winning. It felt like all (or most of) the justification we needed. It set us on a path of building a bigger product than we should have, and committing (falsely) to our own assumptions of what would work, without fully testing them.

Having said that, I also realized at some point in the process – especially as a result of the recession – that in order to survive we would have needed a lot more money and a lot more time. I estimated about $5 million and 3 more years. That wasn't going to happen. But as we got deeper into the market we recognized that to punch a real hole in it would require lots and lots of capital. So while we raised too much, too early, we also weren't in a position to raise a lot more, later when it would have made more sense.

Raised Money Over Too Long a Period of Time

Although I did raise too much money too early, it also took too long. The money was raised in two major tranches, but we also left room for key angel investors to come in at various stages. The entire process took 9-10 months at least. As a result, I spent too much time raising money, which was a complete distraction from running the business. That's the double-edged sword of fundraising — you have to do it (and it's hard) + it's distracting and in some ways detrimental to running your startup. I saw raising money as a form of validation, and so I was willing to invest in it. But it's really only validation that someone will write you a check; not that you've got a strong, scalable business. The challenge with raising money is that once you start, it's hard to stop. The money is used to grow your company, which often results in more expenses. More expenses, without revenue to support them, means going back to raise more money. So you end up on this conveyor belt of raising money, which is difficult to get off of.

Investor Management

I had a great group of investors. This included Montreal-based venture firms and angel investors in Montreal, New York and elsewhere. Unfortunately I didn't manage and leverage my investors enough. From a leverage perspective, I didn't do enough to extract value from my investors. I didn't engage them enough for introductions to potential partners and customers. I didn't follow through as frequently as I should have.

I think the key to successfully managing and leveraging investors is consistency. You need to be consistently knocking on their door, asking for help with specific issues, and providing them the ammunition they need to promote your startup. Investors are busy people. Venture capitalists are managing portfolios and chasing new deals. Angel investors are doing the same, and often times have their own startups they're working on. You need to earn mindshare and attention.

On the flip side, you can't listen to everything your investors tell you. They're not infallible. If you disagree with them you need to step up and say so. It's not easy, especially for first-time entrepreneurs, but it's critical. The relationship between investor and entrepreneur isn't parent-child. It's partner-partner.

Go-To Market

When we were getting ready to launch the paying version of Standout Jobs, we decided to pursue a channel partner strategy. This meant providing partners with a white label version of the Standout Jobs platform so they could resell it. In principle it made a lot of sense: they had existing customers, market share, mindshare and experience selling into the HR market. And we had a lot of interest. But looking back it was too early for us to be seeking partners. We didn't know ourselves yet how to sell the product, to who and for how much. Without that information we didn't provide channel partners with enough of a "sale-in-a-box" that they could execute on.

In some ways we tried to offload responsibility of selling our product to our channel partners. It worked, to a degree, but not well enough. And when the recession hit, everyone in the HR / Recruitment space got clobbered. Our partners recoiled and focused on their core, and we were left holding the bag. I still believe there's immense value in leveraging channels for sales, but only after you know how to sell your product effectively; the channels then provide scale.

Big Pivots

At some point in mid-2009 it was very clear that Standout Jobs was in trouble. The recession was still holding a death grip on the economy, we were running out of cash and traction was middling at best. It was a difficult time. I don't remember the exact order of events, but we decided to pivot in a big way. This was an "Odeo into Twitter" moment, where we were prepared to shutdown Standout Jobs as we (and everyone) knew it, and resurface with something completely different (although still relevant and emergent from our experience.) It was an exciting point and truly re-energized the founders and the team. We had some rough ideas, brainstormed furiously and put together a super-basic prototype of what we envisioned. It was a B2C play around identity, referrals and recommendations. There are a number of startups now that are playing in that space.

We pitched the concept to our Board of Directors but a decision was made to "stay the course." I have no clue if the new idea would have generated any traction, but I should have fought harder for it. We certainly didn't have much to lose at that point. I don't think many startups really go through this kind of massive, transformational change. It's scary. And inasmuch as people talk about pivoting (although this was more like a "scrap and restart completely new"), I don't know that most entrepreneurs and investors really have the stomach for it.

I also tried a few smaller pivots. I relaunched the website and repackaged the product into a few different offerings. Also added a service layer on top of the product. The goal was to take what we had and try to find a market. I also wanted to simplify things and focus on lead generation as much as possible. These smaller pivots and market adjustments actually generated positive response, but I didn't have the time or resources to execute on them aggressively enough.

Bullshit & Politics

I'm not a fan of bullshit or politics, but unfortunately these are both realities when running a startup – particularly one that's not performing well. You have to be prepared for it. I'd recommend having one or two very strong outside advisors – experienced entrepreneurs that have "been there, done that" – to lean on when you get to this stage.

Creating Value

In October, 2009 we let go the bulk of the team. We held on for as long as we could because we were engaged with potential acquirers and felt that the team would add value in an acquisition. And there were a couple of acquirers where I think this was legitimately true. Unfortunately, they didn't see it that way! Eventually I made the decision to let the team go. It's a shitty experience, but not as bad as you might think.

Throughout the acquisition process we worked to create value. The team. Our partners. The technology. Us (the founders.) It's a very similar experience to raising early money; often you don't have a lot of proof of anything, but you try and tell a good story. And fundamentally, I believed at the time (and still do today) that we were doing something right in terms of where recruitment marketing and branding was headed. The package wasn't quite there, but combined with other pieces of the proverbial puzzle and I saw the potential for strong value to emerge.

Perseverance

With all of the mistakes I made and all the difficulties faced by Standout Jobs, I'm still pleased that we (and I) persevered. From the fall of 2009 until the acquisition closed, Standout Jobs was really just me looking to get a deal done. Sort of a "last man standing Ultimately it was too little, too late, but I'm still glad I went through it.

Having said that, I think founders have to be very cognizant of their startup's lifecycle. When it's time, put a bullet in it and move on. Whenever I had the "should we kill it?" conversation with my founders, investors or Board of Directors I always felt that I could extract more value. And I wanted to try. The experience was worthwhile.

The Future

I'm no longer involved with Standout Jobs. The acquirer has been working for some time on the technology and I look forward to future announcements from them. I'm still actively looking at the HR / Recruitment space — frankly, I feel like there's unfinished business there. A lot of entrepreneurs have reached out to solicit my feedback on their HR / Recruitment startups. I'd encourage you to keep doing so. I'm looking for the winners in there; I've gotta believe they're somewhere!

I've now moved on to Year One Labs, an early stage seed accelerator based in Montreal. So I'm (sort of) on the other side of the table, although I can't ever really picture myself as an investor. I see myself more like a guy that wants to help other entrepreneurs build successful businesses. Not because I've sold a company for $100M yet, but because I've been through a lot, seen a lot, and like to roll up my sleeves and get dirty. And at some point in the future I will found another startup (or two, or three, or four.) Once you start, it's hard to stop. Even with the ups and downs, bumps and bruises. I'd rather not repeat the same experience I had with Standout Jobs (at least not in its entirety), but I'd take this life and these experiences over a "regular job" any day.

Original article here: http://www.instigatorblog.com/postmortem-analysis-of-standout-jobs/2010/10/05/

Startup Failure: Cusoy

Cusoy: A Postmortem

It's hard to believe that it's been exactly six months since I first penned my post What can you do in six months? that began this startup journey in the first place (see Six Month Challenge for a table of contents of my past weekly updates).

This is the conclusion of the hardest yet most fulfilling six months of my life, so far.

Today I'm going to write a post-mortem about my startup called Cusoy, a curated restaurant finder for people who are gluten-free (from gluten intolerance, celiac disease or as a healthy lifestyle choice) and/or have food allergies.

I. Why I started Cusoy II. What I accomplished III. Why I shut it down (alternatively, what I would've needed to continue)

I. WHY I STARTED CUSOY

I've always wanted to start my own company and saw everything up until this point as preparation for this dream.

Five key reasons why I started Cusoy:

It solved a personal pain point. It came at a time of excellent market timing and opportunity. It resonated with its intended demographic of users. It came during a time when I had the financial and life means to pursue it.

It originated from a cause I deeply believe in: health and wellness.

It solved a personal pain point.

I am gluten intolerant and this was a big pain point for me of finding "safe" restaurants when I went out to eat with family and friends, and more importantly, beyond "safe" restaurants, "safe" menu items I could actually eat without getting sick afterwards.

I didn't find a good solution out there outside of a variety of solutions: calling the restaurant, asking the waiter/kitchen manager, scrolling Yelp to find one or two comments (if any) mentioning gluten free or Googling. I thought, there must be a better way.

Yes, Yelp has a tag for "gluten free" but it only has 44 restaurants in San Francisco with that tag (do it yourself if you don't believe me and NYC actually has double the number of restaurants tagged gluten-free), which is a very low and deceiving number.

More importantly, it does a terrible job of giving people like me actual relevant information for gluten free: Does the restaurant train its staff? Do they have separate cutters for their gluten free pizza? Are they flexible and accommodating? Do they have a separate gluten free menu? What ingredients do they use? etc.

Moreover, if a restaurant is gluten free and if I am gluten free, dairy free and soy free, what exactly can I eat? What do other people with dietary restrictions like me recommend? I wanted to collect all of that information.

Gluten-free apps out there now like Find Me Gluten Free are just okay and barely adequate, but still don't give me relevant detailed information like I mentioned above. It's almost completely useless to me — my own research is much more valuable.

At the same time, the difficulty lies in having accurate menu information and getting information (that I mentioned above) in a way that, while it doesn't scale, is extremely valuable to the user.

It came at a time of excellent market timing and opportunity.

There are two groups of users: those who have to eat gluten free because of gluten intolerance or celiac disease, and those who choose to do so as a healthy lifestyle choice — with both groups high in number and growing.

In January 2013, the NPD Group released findings that 1 in 3 Americans are trying to cut down or avoid gluten (over 100+ million Americans). In 2012 alone, over 200 million restaurant visits occurred with consumers ordering food described on the menu as gluten-free.

Moreover, there are 3 million people in the U.S. who have celiac disease, 18 million estimated to have gluten intolerance (though many do not know it) and 15 million with other related food allergies (dairy, soy, wheat, nuts, etc).

That's a quick snapshot for those who are not familiar with gluten free and may think it's some "small niche" with no opportunity at all. Not to mention that San Francisco and the Bay Area is the perfect starting point — SF is the foodiest city in the country and the Bay Area demographic is extremely health-conscious.

It resonated with its intended demographic of users.

I took a week to validate this idea with my users and consistently reached out to potential users for additional feedback, resulting in over 100+ quality customer research surveys. Many complained about this problem and were happy to hear I was working on a solution to help them.

It came during a time when I had the financial and life means to pursue it.

I am 23, single, no debt or student loans with 6+ months of financial runway. I completely bootstrapped Cusoy from my savings and did not have any other distractions or obligations.

It originated from a cause I deeply believe in: health and wellness.

People don't realize the immense importance of proper diet and nutrition. What you eat affects everything, especially your health (not to mention your energy, productivity, etc). The contribution of your diet to serious medical problems and digestive health is often overlooked in favor of complex procedures and prescribing complicated drugs with a lot of side effects. What if we started helping people at the base of the pyramid of having symptoms, rather than the end-stage of the disease?

II. WHAT I ACCOMPLISHED

Start: July 15, 2013 Shipped: November 8, 2013 End: December 31, 2013

Summary: customer research, product development, user acquisition, usability testing, sales, email marketing, analytics/KPI's, pitching (Rock Health and Greylock Partners), customer support

- Successfully validated a market need that also solved a personal pain point
- Narrowed initial broad focus (vegetarian/vegan, Paleo, gluten free) to owning the gluten free space
- Researched gluten free via books, medical lectures and conversations with healthcare professionals
- Collected 100+ quality customer research surveys
- Sourced 80+ beta testers through organic user acquisition
- Designed a 3-week email experiment to study user interest vs. intent to purchase
- Built 3 MVP iterations despite being non-technical
- Product activation (sign-up & meaningful action) in first 30 days: 50%
- Product retention in first 30 days: 25%
- Conducted 10+ in-person usability tests in San Francisco/Greater Bay Area
- Scheduled 6 meetings with Rock Health over the course of 5 months for product/application feedback
- Invited to pitch practice with Greylock Partners and learned how to best pitch Cusoy for seed funding
- Consulted an informal advisory board of current tech employees, former and present entrepreneurs

I felt that Cusoy did succeed in solving the initial problem I set out to solve, but ultimately it was difficult for me to generate money from it (or at least see a clear, predictable way to sustainability).

III. WHY I SHUT IT DOWN (ALTERNATIVELY, WHAT I WOULD'VE NEEDED TO CONTINUE)

Four reasons (in order of least to most important):

No team No funding No longer aligned with my personal goals anymore No clear or predictable way to sustainability (B2C is extremely, extremely hard to monetize) No team

It's really hard to do a startup by yourself, especially if you're a non-technical founder. No kidding, right? However, being non-technical wasn't my excuse since WordPress came to the rescue and I basically bought the development through using cost-effective themes.

Rather, it was overwhelming both physically, mentally and emotionally — morale wise, it was incredibly exhausting. Without anyone else to keep me accountable or really caring about Cusoy as much as I did, it was hard to deal with all my self-doubt and still persevere on when I was essentially my own cheerleader.

I was juggling many things at once — including manually adding restaurants (this was my biggest gripe), sourcing new restaurants, talking to restaurants and users, sourcing new users, reaching out to restaurants, etc. It was just too much work for myself, with little hopes of generating even a dollar in revenue.

At the very least, I would've liked a sales/marketing/business cofounder who would've helped to hustle with restaurant outreach, content marketing and so forth. At this point, I didn't need a technical cofounder since I could handle it myself (for the time being).

What about trying to find other people?

I tried Meetup groups and a couple Hacker and Founder lunches and even met another founder working on a food-related startup. I had a meeting with a potential front-end developer from Reddit, and then also met with a potential user from Meetup who actually made a gluten free app four years ago but shut it down. Ultimately, over time, these meetings didn't pan out.

Over time, I realized the right match would've been a developer who also was facing problems eating out while gluten-free.

I also realized finding a technical cofounder is not just trying to find a cofounder that matches you in terms of complementary skills (do they know Ruby on Rails? node.js? etc), but also having the same values, vision, work style, expectations, etc. It's exactly what you've heard — akin to a sexless marriage with the startup as the baby of the relationship.

Moreover, it's difficult to compete with sky-high salaries and perks engineers already enjoy from working at Google, Twitter, Facebook, etc. when I couldn't offer a salary, funding or even also strong numbers of product/market fit (or better yet, paying customers or revenues of some sort). I didn't come across as many who faced this sort of problem of eating out while gluten free, so it was difficult for them to understand and thus find it an interesting project. Not to mention everyone seems to be working on their own projects as well.

I didn't expect this to happen overnight but went back and forth in terms of how much time I was "wasting" trying to meet other people vs. working on the product itself.

My hypothesis of five metrics of finding a technical cofounder

I was thinking about this and after talking to some engineer friends, came up with five metrics that cofounders (especially technical ones) care about when they think about partnering up with a non-technical founder.

These are not in any particular order:

- Is the founder technical (or at least trying to learn how to code?); the founder's background/domain expertise Product/market fit, validated idea, paying customers, etc
- If the project is personally interesting / challenging to them
- Equity / salary / compensation
- Similar vision, values, work styles

I'm definitely keeping this in mind as one of my goals for 2014 is to find a technical cofounder for my next venture, this time in B2B SaaS.

No funding

Since Cusoy is not revenue-first, I was completely bootstrapping from my savings and it was difficult to keep going without watching my money dwindle without much to show for it.

What about trying to apply for funding or contacting investors?

I did, with Rock Health. I went to six different office hour sessions over five months with them and applied to their most recent batch, but unfortunately did not get in due to their concerns that Cusoy's nature would be very difficult to scale and grow and that I needed another cofounder to help shoulder the work. Not to mention Rock Health is extremely competitive, with 99% of their portfolio being B2B enterprise companies, and hardly any B2C ones like Cusoy.

I could've tried to reach out to angel investors who are personally interested in the food / health and wellness space, but again, since I was the one doing everything, I didn't see trying to get funding as a productive use of my time (outside of working on the product and talking to users). Apparently, a rule of thumb suggests that 1/60 actually invest.

To make it worthwhile to look for funding meant I would need strong growth/traction numbers (like 10%+ MoM) that proved product-market fit to make a compelling case, which I would need more manpower help to get (see: no team). It was kind of a self-perpetuating problem.

I had great early traction with my beta testers very early on, but I felt really burned out and demotivated shortly thereafter… not because Cusoy wasn't doing well, but that I had increasingly come to the realization that it just didn't fit with my personal goals anymore and left me physically and emotionally exhausted.

No longer aligned with my personal goals

Over time, I realized Cusoy is not meant to be a small-scale, one-person venture.

I don't know what I expected going in (certainly not everything that has transpired since), but for some reason I naively thought the money part would take care of itself and that somehow by some magical force of willpower, Cusoy would become self-sustaining without the need for investors.

Wrong.

That's when reality hit, a couple months ago.

I wasn't so much concerned that Cusoy's growth didn't skyrocket to thousands or hundreds of thousands of users, but more so that it was indeed providing value and that I could perhaps take it to a point where it would become ramen profitable and I could just organically grow the product, team etc from there.

Cusoy has the incredible potential to become an organically grown gluten-free brand and community, not simply just a curated restaurant finder. Once I owned the gluten free space, I would have expanded to vegetarians and vegans, Paleo and the like. You can get an idea of how big Cusoy could get (dream big, right?) and particularly given the perfect market timing and opportunity right now.

What were my personal goals and how did they change over time?

In the beginning, I wanted to build a beautifully designed product with a great user experience that helped solve a personal problem, but more importantly, also a problem other users had.

Over time, I realized I wanted to build a beautifully designed product with a great user experience that helped solve a personal problem for me and for other users… that also generates revenue.

I didn't want a startup, but an actual business that generates revenue, and Cusoy would not fulfill that personal goal for me without a full-time team, 1-2+ years of funding, multiple years of hard work (3-5+ years at the very least?) trying to answer the if/when questions of whether or not Cusoy could make money (very expensive questions too, might I add — not only in money but time, my most valuable asset).

While I know there might be a possibility I could hustle incredibly hard and try to set up partnerships, the time investment required far outweighed the already incredibly slim chances of generating revenue.

Cusoy ultimately no longer aligned with my personal goals.

No clear or predictable way to sustainability (B2C is extremely, extremely hard to monetize)

This is probably the #1 reason that really was the straw that broke the camel's back.

I left the most important reason for last.

If I didn't have a team or funding but had clear and predictable ways to sustainability, I would have persevered with Cusoy. Absolutely, no holds barred.

Back in October, I read this TC piece on how the percentage of free apps in the App Store will grow to 91-93%. Wow.

Actually, as recently as two days ago, Gartner predicted that 94.5% of apps will be free by 2017.

It makes total sense — if you think about the increasing consumer expectations of free apps and free content. It's incredibly hard to convince a user to pay even $0.99 (less than a cup of coffee) for an app, since there are many free alternatives available that make it difficult to overcome the friction for consumers to pay.

This posed an interesting dilemma since I was both the developer and also a user myself of Cusoy, and I certainly would not have paid to use Cusoy. Cusoy is too similar to Yelp, and consumers don't pay to use Yelp, they don't even need an account to use Yelp. Obviously, Yelp charges the businesses in order to make money.

What about charging restaurants? Partnering with OpenTable and taking a cut of reservations, etc?

Of course, I thought about this, but without strong user numbers and growth to prove to restaurants that Cusoy was in fact responsible for 10 extra reservations at XYZ restaurant last Friday night, it would be hard to convince them to advertise with Cusoy.

Bottom line: I needed strong user growth numbers and concrete evidence before I would have leverage to use when talking to restaurants to advertise and partner with Cusoy. I couldn't get those numbers yet working by myself (especially with no team or funding).

I even played with a certain pricing model — of $1-9/month pay-as-you-go and at-your-discretion payment plan. For instance, if one month you decided to try cooking yourself at home, it may be worth $1 to you a certain month. Then another month, if you're traveling to San Francisco and are not familiar with the restaurants that are gluten free, you might use it 5 times over the course of a week — and it may be worth $9 to you (a tiny cost at the expected value of not getting sick).

But even that was an uphill battle and left a sour taste in my mouth — is it right to charge users when they are helping to generate content for Cusoy?

What is the crux here?

Monetizing user-generated content is really, really hard. Especially for Cusoy, where it heavily relies on users to crowdsource information and generate new restaurant additions, it doesn't encourage or incentivize users to do so if they have to pay to use Cusoy.

It's interesting too — because it's kind of a brute force, numbers game with Cusoy. If you think about it, Yelp has 117 million monthly unique visitors but I'd wager only 1% of users actually post reviews, and just a slightly higher percentage actually register as users (not sure how many active registered users there are, though).

With Cusoy, it is powered by my manual research and vetting of restaurants but heavily relies on active and registered users to post reviews, rate restaurants and even submit restaurants to list.

I would need incredibly high numbers of users since only 1% of them would actually register and contribute back to Cusoy rather than passively consume its content. Again, this goes back to needing another cofounder, team and even funding to make this a reality.

Why is it so hard to monetize B2C startups?

This past fall 2013, it was also painful to see a lot of B2C startups shut down — from Forkly, Everpix and Sonar to Flud, Cue and Tutorspree. They shut down for various reasons, but mainly either not being able to reach product/market fit or becoming profitable.

I got a taste of it myself with Cusoy as my personal experience.

B2C is really, really hard to monetize and thus hard to succeed in the long run. Unless you can find a big enough pain point — or go into a deep niche problem, it's very tough to convince users to pay you.

In contrast, the inconvenient truth is: if you run a B2B startup you're more likely to succeed.

Overall, Cusoy served as a great lesson in startups for me and I don't regret doing it for a second, in spite of its failure to take off or generate revenue.

Original article here: http://melissatsang.com/2014/01/15/cusoy-a-postmortem/

Startup Failure: FormSpring

Formspring - A Postmortem

This Sunday, after nearly three and a half years, Formspring is shutting down. While it's a sad moment for me to see a startup I worked on and loved close its doors, I can't help but also look back and examine ways we could have steered the product toward a more successful outcome. What follows are a few points that I've been reflecting on ever since Ade announced Formspring's closure that I hope will be helpful for current and future entrepreneurs. To be clear, my critique is of product decisions we all made together, as a company (while I was there). As we would have shared in its success, so too shall we share and learn from its end.

We protected anonymous content to a fault

Formspring's initial success was, in large part, due to giving our users the ability to ask each other questions anonymously (even without a Formspring account). In under a year, we skyrocketed to our first billion questions answered and showed few signs of slowing down. Yet even as we celebrated these milestones, we were all discussing how anonymity would or wouldn't play a part in the future of our product. On the one hand, anonymity was a really popular feature (duh). On the other hand, we saw a lot of bad and abusive content come through that channel (double duh). A fact that we wound up being pretty infamous for.

But man was it hard to let go of anonymity as a core feature. We tried workaround after workaround. We prompted for sign-up after asking an anonymous question. We started pushing privacy settings for users into our on-boarding (which they never changed, of course). We started setting up elaborate filters to catch bad or abusive questions and put them behind a "Flagged Questions" link in users' inboxes.

We spent a lot of time on anonymity. It was our sacred cow. Looking back, we should have spent that time finding ways to gracefully degrade that feature instead of finding ways to keep it alive. When you find yourself constantly giving a feature CPR, you should stop and consider whether or not it's worth saving (or even possible to save).

Our opaque follow-model shot us in the foot

In a way, this lines up with our stance on anonymity. Following on Formspring was, for years completely anonymous. You couldn't see who followed you and others couldn't see that you were following them. This meant that we gave people a microphone and they kind of had to hope people heard what they were saying. And until we eventually launched our Smiles feature (akin to Facebook Likes), there was no way to know that your content was being consumed. We debated this a lot internally and came to the conclusion that the Twitter public-follow model was broken in that it put unnecessary social pressure on users to follow back. We felt we could build social features on top of the content (like Smiles) that let our users receive feedback and let their followers out themselves purposefully.

Formspring eventually allowed public following (not as a default, and after I left), but it was too little too late. My takeaway from this has been to always double check to make sure you're not designing toward your own biases instead of what's best for your product and users. Formspring had clearly struck a chord with people aching to share more about themselves with their friends. And instead of making it apparent that they were achieving their goal, we put an artificial barrier in place and prevented them from knowing if Formspring was working for them or not.

We skated toward the hockeystick

The biggest sin of them all from a product perspective, but also the hardest to avoid (and one that I see companies make over and over again).

Our initial graphs at Formspring, as you probably know, all hockeysticked up and to the right. Nearly straight up. That part was totally awesome! We were super popular! We could be the next [insert gigantic company name here]!

Oh wait, the graph has peaked and is starting to slowly (very slowly) trend downward. What do we do? Make big bets, right? Try to recapture that crazy growth!

And so we tried. The first big project we worked on was a Formspring button that sites could embed at the end of blog

posts or other content. We had millions of users, so we figured it wasn't a stretch to imagine they browsed other web sites and would gladly click a Formspring button at the end of a post (which asked "What did you think?" and allowed them to post a response to their Formspring page). This was just as the Facebook Share and Twitter "Tweet This" buttons were appearing, so we figured it made perfect sense to follow who we viewed as our closest competitors at the time.

We literally spent months on that system. We had to make sure our servers could handle a potentially huge influx of traffic (we based our estimations on our main site's traffic, which was honestly insane), had to design and implement the feature, make sure the implementation was easy for publishers, make deals with publishers, etc. We bet huge. On someone else's (Facebook and Twitter's) plan.

Flop.

How about photos? Instagram had just come out and was killing it! Our users would surely be more engaged if they could represent their questions and answers visually, right? I worked on those designs for at least a month and a couple more were spent iterating on and actually implementing that work.

I actually don't know exactly how that launch went (because I had moved on to Amazon right before it went out the door), but now it seems obvious that photos weren't the killer feature we'd all hoped they would be. Oof.

Entrepreneurs: build your product, not someone else's. The most successful products execute on a vision that aligns with their product's and users' goals. It's hard to put blinders on when your stats are slowly coming down and you see other startups skyrocketing around you with various tactics and strategies. For the love of god, put them on. It's the only way to build what you should instead of chasing others' ideas.

Also, print out Paul Graham's Startup Curve and put it on every wall in your office. It will be you. Buckle up.

~

Even in the face of Formspring's shutdown, I'm grateful for having been a part of it. I learned so much from both the team we built and the product that I'll carry with me the rest of my career. I learned that small changes can frequently (and surprisingly) have the biggest impacts. I learned that gigantic, sweeping changes make it hard to track what's going right and what's going wrong. I learned just how hard it is to keep a huge site up and operational (also what happens when all of EC2-East goes down). I learned that I love designing for people I can communicate with every day within the product.

But most of all, I learned to focus. To put the product vision first. To not try keeping up with what's trending. To build your product for your users.

Thank you, Formspring. It was a wild ride.

Original article here: http://blog.capwatkins.com/formspring-a-postmortem

Startup Failure: Mass-customized Jeans

Internet Startup: Lessons from failure

Is mass-customisation a viable business model?

'The devil makes work for idle hands', my grandmother was fond of quoting. Loving my granny and fearing the devil, it's no surprise I developed an irrational urge to be busy at all costs. Chronic busy-ness and obstinate determination are a dangerous combination.

How dangerous? Enough to make me start a mass-customised denim label offering 12 million variations at launch. And fail hard.

The idea was a website that let you design a pair of jeans, from the cut to the colour and everything in between, and have them delivered within two weeks.

Partnering with a uni friend, we pitched our concept to a whole bunch of web developers we knew. The challenge we posed was that it couldn't be customisation by drop-down list or radio buttons (solutions we were offered repeatedly). This was 2005. We needed to show customers high-quality photographic customisation, where every selection changes your product. Custom-made product required a strict no-refund policy, so customers needed to see exactly what they were going to get.

All of them said the site we wanted was impossible, apart from one who quoted upward of $40k for build, excluding the photography and image editing that would be needed. At this stage we only had $20k total capital for the entire venture —sampling, pattern making, inventory and production.

So my co-founder taught herself to code. She worked day and night—we did all the operational planning during business hours, and when I went off to a night job managing a restaurant, she'd code, with Sex and the City for company, until 1 or 2am when I stumbled in again from night shift.

We went live in August 2005 with 12 million variations on offer for both men and women—5 cuts each, 10 varying weights and colours of denim, as many wash options; gold, silver and bronze rivets; a choice between button or zipper fly; several thread colours, pockets and knee patches and belt loops, quilting, lining, printing and any length you could ask for. 12 million variations. Insane.

We didn't validate our hypothesis.

Our hypothesis was "people want custom-made jeans".

We didn't think of it in terms of hypothesis. We just thought this would be a cool business and maybe we could make a name as designers. We never considered ourselves an internet startup. We were, but we didn't see ourselves that way.

Why did we launch with 12 million variants? Because we thought we needed to. Would the market have been happy to customise just the length? If we'd done proper validation, we would have known, that yes, it would have been happy—thrilled even.

It only became apparent, too late, that 90% of our orders were for very typical jeans from atypically tall women. Stonewashed stovepipes with a 36-inch leg, anyone?

Pricing our products for less than free.

We were retailing at $200 a pair, about 50% more than off-the-rack Levi's, and about 50-200% less than off-the-rack boutique labels. According to our research, $200 was the price the market was willing to bear for such a high-risk purchase—jeans from a new brand with no reputation, no physical presence to try them on, and no refund policy.

Our factories—both the makers and denim laundry (who "wash" the jeans after sewing)—agreed to produce single units without the usual promise of a bulk follow-up order. And that only because we had a good working history with them. A

finished unit came in at around $50, high but bearable. Remember bulk units would have been single digits.

We chose not to factor in our own labour as a production cost to keep our retail price artificially low while entering the market. Administration of orders, the time-consuming cut-and-bundle process (which we did ourselves) and weekly trips to the factory added up to around nine hours per unit. Let's say we priced our labour at minimum wage, like I was getting at the restaurant, at $15/hour. A single unit would have shipped at a cost of exactly $200 (yep, our retail price).

Fortunately we didn't have overheads. We were living together in a double story terrace and used the ground floor as our studio, so we shouldered rent and utility costs personally. As a result, our shipped cost was more like $60.

We weren't going to draw from the business until we had recouped our (parents') initial investment. That meant continuing to operate 9-5 while earning an income at night, which was fine for the months leading up to launch, but totally unsustainable once orders started coming in.

A difficult choice

We were exhausted, paying for the privilege of making jeans. Our seed money, plus another $10k, bringing total investment to $30k, was all but squandered.

We had to decide on the future of the business. We didn't want to open up a retail space because we'd just be localizing our distribution (with bonus overheads!). But now we'd tasted independence, developed heroic determination and suffered grandiose fantasies of stockists falling over their Manolos to buy what we were selling.

So we decided to wholesale, on the condition we could establish a cost-effective supply chain. This might all seem like a juvenile, ill-informed, poorly researched, gamble. And it was mostly, but I heard an irrepressible voice day and night,asking what if it works?

We added a line of jersey basics to our denim line, packed up our lives, ended the lease on our house, loaded our portfolios and patterns into suitcases and boarded a plane for Malaysia, where I had some contacts in the manufacturing industry.

At one factory after another, we were wooed by the boss, we got the grand tour, we were asked our volume projections…. and we were shown the door.

Our (very inflated) annual forecast of 5000 units per style was daily output for these guys.

My contacts, as it turned out, had developed the licensed merchandise for MTV, and their contacts in turn, expected MTV volumes. The single factory willing to work with us said they'd quote on our items while showing us around the facility. When our guide accidentally led us full-circle, we caught their product team copying our patterns and that was the end of our sourcing trip.

Is mass-customisation a viable model?

It's possible that some of the big luxury houses have developed a profitable mass-customisation model. But in my opinion, only because the "customisation" component is not much of a divergence from their existing "mass" production process. Adding your initials to Louis Vuitton luggage for example—where hand finished, if not entirely hand-made product, is quid pro quo and the price premium is not an issue for the existing, money's-no-object customer. If a $10k handbag is just another handbag, what's another few grand for a personal touch, right?

And at the low-margin, high-volume end of the market, you could leverage economies of scale built into an existing supply chain. I'm thinking of Nike's custom sneaker offering, NikeiD. I'm not claiming to know anything about Nike's financials, but even if they run the custom product offer at a loss, it's a unique kind of brand marketing they wouldn't be able to achieve—let alone buy—with traditional advertising.

As a new business, without either a handcrafted mass line or a high-volume, latent supply of components, mass-customisation, as a model for discretionary consumer products, is a difficult proposition.

What could I have done differently?

If I were to launch this business today, 2013, instead of relying on qualitative data I would:

Set up simple landing pages prior to building even MVP to validate concept

Leverage the social media tools now available (for context, we launched about a year before Facebook)

Pitch the concept to established brands, retailers, entrepreneurs and investors with deep knowledge and experience in this market

Look outside of denim, to products like business shirts, or better, ties—which don't require fitting, and beyond fashion to other consumer product categories

Lastly, I'd consider the customisation itself, rather than the jeans, as the product—we could have customised anything with that website. I would look for ways to scale that IP.

Determination is dangerous

We're conditioned to believe that "obstacles are what you see when you take your eyes off your goal", so "never give up" and "follow your passion".

Determination, big dreams, and old-fashioned inexperience kept me from learning three things that would have changed everything:

Validate

Launch small

Iterate often

I'm not saying don't be determined. Rather, I'm suggesting you replace those tired old clichés with these: "obstacles are what you see when you refuse to be led by emotion" so "give up often" and "be passionate about creating real value".

Original article here: https://medium.com/@markbarwald/internet-startup-lessons-from-failure-995d0285e8ba

Startup Failure: Parceld

Lessons from my failed startup

I stumbled into founding a company, and perhaps this is why my mistakes seem like "DUH" ones to others, but it's also why I was willing to take risks more seasoned entrepreneurs or business people were too scared to take.

Editor's Note: You may remember that back in April we published an interview with entrepreneur Brianne Garcia. At the time, Garcia was a fellow at the Tow-Knight Center for Entrepreneurial Journalism and working full time on her startup Parceld. Since we last heard from her, Brianne has graduated, won some funding, worked on Parceld, started to raise money, overcome innumerable hurdles, and finally, decided to close Parceld down. In this blog post, Brianne shares with us her lessons learned.

When we entrepreneurs decide to start a company, we know that more things can go "wrong" than can go "right." We understand this, and yet we believe that we have what it takes to put "a ding in the universe", as Steve Jobs once said. Startups are so "cool" right now that it's easy to forget how terribly naive, blindly optimistic and delusional startup founders have to be to think their company will succeed over others, especially over others with similar ideas in the same market.

I stumbled into founding a company, and perhaps this is why my mistakes seem like "DUH" ones to others, but it's also why I was willing to take risks more seasoned entrepreneurs or business people were too scared to take.

At this time last year, I had never stepped foot in the tech community in New York. To say it was foreign territory would imply that I even knew it existed. Instead, I was fully engrossed in finishing my last semester of journalism graduate school, and was anxious about where I'd work upon graduation.

I had always been interested in online communities, and had hoped that I might be able to merge that interest with my lifelong obsession with fashion. Even though I wasn't part of the tech community in the way I am now, I had always been an early adopter of shopping and social sharing sites. I hopped on Polyvore, Svpply, Pinterest and Tumblr very early in each of their respective lives, and that's where the idea for Parceld crept into mind.

As an avid user of these sites, I noticed gaps in the process of being inspired and being able to actually DO something with this inspiration (namely, buy).

My company, Parceld, would solve the problem discovery sites create: the desire to purchase a specific style or trend (since a bunch of images aren't actionable or affordable). Parceld would let women tag photos to say "here's what I like about this look. Show me something like this", and we'd surface a pool of suggestions, sourced from some key major brands as well as emerging brands and retailers.

When I graduated, I had a choice: go get a job at a media company, or try to make Parceld a business.

I chose the latter, and am so glad I did. Even if, after 7 months of dedicating myself body, mind, and spirit to the effort, I have now shut Parceld down.

Two months ago, I stood at another crossroads: I had run out of funding, my technical co-founder had quit and pulled the code out from under me, I was out of savings and student loan payments were taunting me from around the corner.

My gracious network of friends and advisors gave me advice that ranged from "keep on keeping on" to "go find a job, NOW." One person told me to try bartending while I figured out next step.

I was pointed in the direction of this farewell-of-sorts blog post from Buyosphere founder Tara Hunt, who recently decided to take a position elsewhere, and discontinue a 100% focus/dedication to her own company. Her biggest lesson: "it DOES take money to build stuff. And time. And those who have a large supply of both have more runway to make several mistakes on the road to über success."

Most of us don't have big wads of cash and time to burn, so we have one shot and then we have to figure out how to pay the rent and feed ourselves. Those who achieve success in that one shot are just as lucky as they are admirable. And

those who don't hit it big on the first try, but have the time and money to figure stuff out, are extremely privileged.

Though it's natural to make mistakes, it's also worth reflecting on those mistakes in order to do better in the next go-around. I made mistakes, BIG ones, that directly contributed to ParceId failing.

With that in mind, here are my top 3 lessons learned:

LEARN HOW TO CODE, and USE GITHUB

Obviously, this is stressed and discussed every which way we turn on the internet, and for good reason. But this is probably the #1 "complaint" I had while building ParceId: I felt helpless in some ways because I couldn't actually code.

In retrospect, I see that you don't have to become your own CTO, but knowing what's going on under the hood can save you tears, money and time. My team and I started using Dropbox before switching to Github, and it was too late for some of the code. If you're a founder, you probably HATE feeling helpless, and when I went to visit the alpha product and it wasn't there, I broke down. Feeling helpless sucks.

The second I get some dough in the bank, I'm taking some courses beyond my Codeacademy online sessions, which are fun, but not really very actionable.

BE AGGRESSIVE

No one likes someone who is too aggressive, but looking back, my idea of "too aggressive" could probably fit very nicely into the "persistent" bucket, which, quite frankly, is not enough when raising money. My father told me that, especially as a woman, to never be afraid to ask for what I want or to remind others of their commitments. People these days are busy, forgetful and over-scheduled; it's quite possible my three emails each got buried, so a fourth or fifth email (not daily, though; maybe weekly) would have served me well. I'll never know.

ASK FOR HELP.

Sure, I asked for money from my family and help from a few friends, but there are plenty of strangers I could have tapped into that could have added immense value to either the product, my time management, or other major aspects of my biz. I say "could have" because I'll never know. Ask for help when you need it, accomplish more. Easier said than done, I know. But always worth a try.

So what's next for me? The truth is: I'm still figuring it out. All I know is this: hustle never dies, it just changes shape.

Original article here: http://skillcrush.com/2012/11/14/lessons-from-my-failed-startup-parceId/

Startup Failure: Saaspire

Lessons From My Failed Startup

Two years ago, I signed on as the third partner at a company called Saaspire. At the time, I had two goals: to build a sustainable business and to learn. The business part didn't work out too well, but I did learn a quite a bit.

Conflict is good…in moderation

One of the things that made life at Saaspire interesting was that my partners and I often had differing opinions. Many times this was a good thing. For example, when working through the design of a new feature or product we would often come up with very different approaches which given time and discussion we could merge and mold into a final solution that was better than anything that we had individually come up with.

Over time however, we started to have larger and larger disagreements. Most significantly, our disagreements expanded scope from product decisions to questions of direction for the company. Eventually we decided to split up and go our separate ways. One partner left to pursue his own endeavor while my remaining partner and I continued on with the Saaspire name. All things considered, the separation went smoothly and everybody left on good terms. That being said, the whole process and the disagreements leading up to it were a huge distraction that we sunk an incredible amount of time and money into.

Lesson learned: Make sure everybody is on the same page in regard to the company's direction as early as possible. What do I mean by direction? Are you going to bootstrap the company or try to raise funding? How ambitious of a product are you looking to build? Are you going to focus on one product or create a suite of products? Obviously you can't know all of the answers to these right away and things will change over time, but the sooner you can work them out, the happier everybody will be. Leaving any of these questions open for too long, especially due to an internal disagreement, will only burn up resources you don't have and distract you from building your business.

Focus

One of the main points of contention that lead to our split was the question of focus. My partner who left, envisioned a suite of interoperating products built on a common platform. Initially this seemed like a great idea as it allowed us to play with different ideas and target different markets while leveraging a few core technologies.

In practice however this proved to be problematic. We were a three person company bootstrapping our products by doing consulting work. We didn't have anywhere near the bandwidth to execute on this type of plan. Instead of having the benefits we envisioned, the platform created a massive overhead on our development work. We ended up over engineering our systems so they could support both today's and tomorrow's products. Similarly we ran into all kinds of weird user experience problems. We would try to design our initial application so that it would make sense in the context of the other applications that we had planned. Unfortunately, in the mean time the app seemed confusing and overly complicated to users who didn't know or care about the planned suite.

After the split, my remaining partner and I relentlessly cut down the scope of the product. After doing so, so many things became easier.

Before, if we gave somebody our elevator pitch they had no idea what our product did and we couldn't really blame them. After, we could give the pitch, people would get it right away and they'd be really excited about it.

Before, our product iterations would take weeks if not months. It was virtually impossible for us to really learn from user feedback. After, we could iterate in days and we radically changed the product (for the better) several times based on user feedback.

Before, we couldn't even sell our own consulting clients on really using our product. After, not only were our clients sold on using our product, we were getting traction with the outside world.

On virtually every level, the more we could limit our scope and focus, the better things worked.

Lesson Learned: Focus. Cut things down as much as you possibly can and then cut some more. There is so much you need to pay attention to and get right when creating a new business and a new product, the last thing you need to do is add to that list. The more you can cut things down, the easier everything will be to test, communicate, sell, and improve.

If you're bootstrapping, play the long game

As I mentioned, the plan with Saaspire was to pay the bills with consulting work until we could sustain ourselves from product revenue or until we could raise a round of funding. While this approach has its merits, it is fantastically difficult to do well.

For a while this model worked pretty well for us. We had one or two long-term clients who we spent a bit less than half of our time working with and the rest of our time was mostly spent on product work. At times this was a frustrating setup in that progress on the product was painfully slow, but it worked and it was stable for a time.

Eventually, as always happens in consulting, our long-term client work dried up. We had saved up a decent amount from that consulting work so we didn't need to urgently get new work in the door, but we did have to make a decision on what to do.

At the time we decided to spend a little time going after some small, short-term work to get some cash in the door but otherwise spend most of our time working on the product and trying to raise a round of angel/venture funding to allow us to work on product full time.

Ultimately this proved to be a fatal decision for the business. We went after funding because we needed it and not because it was the right time to go after funding. If we had instead spent the time sunk into fundraising into selling new long term clients and into growing the consulting business into something a bit more sustainable, we probably could have bootstrapped the product to eventually be self-sufficient if not profitable.

Lesson Learned: If you're bootstraping, cashflow is king. If you want to possibly build a product while your revenue is coming from other sources, you have to get those sources stable before you can focus on the product.

Sometimes you need to do the wrong thing

We actually saw this bootstrapping problem coming way before it happened. We had plenty of time to correct course and focus our energy on building a stable consulting business before building the product.

Why didn't we do it? We really didn't want to. It was one of those situations where we had to decide what we wanted in life.

If our only priority in life was building a successful, independent business, building out a larger, more stable, consulting business absolutely would have been the right path to go down.

Unfortunately, at that point both my partner and I were pretty burned out on the consulting game. While we had the skills needed to build a larger agency, we knew we would have been miserable doing so. It simply wasn't something we wanted to invest the next two or three years of our lives doing.

Given the choice of building a larger consulting business and going back to "normal" jobs working for the man, the latter was much more appealing to us.

Accordingly, we made a try at raising a round of funding as we felt we had a small chance at making it work. In the worst case scenario we'd still be in the position of shutting the business down but we would have learned a bunch and we'd be in a place were we'd feel we'd given it our all.

In the end, even though trying to raise a round of fundraising instead of continuing to bootstrap was fatal decision for the company, it is not one I regret.

Lesson Learned: What the "right thing to do" is and what the "right thing for you to do" are not always the same thing. Be honest with yourself and your partners about what you want and what will make you happy.

If you're raising funding, beware of the valley

In this case I'm not talking about the place on the left coast but instead, the dangerous place between launch and traction.

In our effort to raise a round of funding, we made several mistakes. The biggest of which was to launch a product without being ready to fund it for however long it would take to show interesting levels of traction (e.g. people paying for our product).

In the investment game you can sell people on a story until you launch your product. Pre-launch you can potentially plant the seeds of something amazing and let people's imagination run wild. Post-launch, you have something that knocks down that imagination back down to the ground, data. Once you have data, you have to be able to survive until you have something based in reality that is exciting for an investor. In our case, we probably would have needed to show low customer acquisition costs and high retention rates. In other cases, you might need to show the ability to form partnerships or collect data.

To be clear, I'm not saying that either approach to fundraising is better or worse than the other. I am saying that you need to clearly fit into one of those categories or be ready to pay the bills until you do.

Lesson Learned: Launch == Reality. Don't launch until you're ready to deal with reality for a while (and then some).

Your product is just a checkbox

This was one of the most painful but also the most significant lessons to come out of my startup experience. You absolutely need to get your product right, but it is just one of several equally important things you need to get right to create a sustainable business.

Even if you ignore the focus issues that we went through, we spent way too much time, energy, and attention on our product. It really hurts me to say that. I was super happy and proud of how our product turned out. It worked pretty well, looked good, was easy to use, and it solved problems for real people. Unfortuantely, there are so many other things that you also need to get right to build a viable business.

In our case our product was good, our business model was reasonable and well worked out, our marketing plan was terrible. It wasn't until pretty late in the game that we really tried to figure out how we'd go about selling our thing to customers. More specifically how would we get in front of our customers in a way that we could afford but would still grab their attention. Given that we were targeting SMBs this is an infamously hard problem and turns out was what really needed to be novel about the business for it to be successful.

To make things worse we also didn't look at the marketing problem until so late in our process that I suspect we would have had to make major changes to both the business plan and product to address the realities of the marketing situation. I say I suspect since we did this so late that we didn't get far enough towards solving the marketing problem before we died to really know.

Lesson Learned: You have a fixed amount of time and attention. If you put too much of it towards any one of the things you need to get right (I'm looking at you product), you will fail at something else.

It was a fun ride

In the end, I look at my startup adventure as my version of grad school.

On the down side it was a ton of work, my relationships with friends and family suffered, and I picked up a non-trivial amount of debit.

On the plus side I learned a ton, I gained new perspective on the world, and I called it quits before any of the downsides became unrecoverable.

Today I'm back into a much more "normal" job working as an engineer at bitly and I love it (we're hiring by the way) . Someday I'll probably make another pass at buiding my own business but for now, it's good to be doing something else.

Startup Failure: GroupSpaces

Looking back at 7 years with my startup GroupSpaces

Today I announced that I'm joining the team at Stripe, after 7 years working on my startup, GroupSpaces. The decision to join Stripe was by no means easy, to say the least. The journey bringing GroupSpaces from a college-room idea has been the best ride of my life - an unbeatable learning experience full of the hardest challenges, the opportunity to work with people I have the utmost respect for, and the creation of a product of which I'm immensely proud. I'm indebted to all the mentors, advisors, investors and all who've supported us along the way.

Going back to when I first started working on the ideas that became GroupSpaces - before we formed a company, lost and gained team members, reformed a new company, took time out of university and raised investment - it's been a 7 year journey for me. 7 years ago, my cofounder David and I knew nothing about business or startups. We had no connections to anything like the ecosystem that exists today in London.

7 years is a long time in any context, particularly for a business that has not become the proverbial billion dollar success. But I'm at one with what Ian from Songkick (who I highly respect for being down to earth in the face of much of the hype and hubris of the startup scene) had to say about being open and proud about the length of time it takes to build our companies. In this manner we fought a long hard graft, assembling the cliche'd airplane on the way down.

Today, GroupSpaces is a sustainable small business, doing that old-fashioned thing of making more money than it spends, and continuing to grow user base and revenue. A fascinating diversity of organisations use GroupSpaces to ease management of their membership, communications and activities - from professional associations to sports clubs, local hobby groups to prominent international campaigning organisations, campus-based student societies to community groups and charities.

While disappointed that the business grew slower than we hoped in financial returns, I'm hugely proud of the contribution and impact we've made, and I remain dedicated to seeing GroupSpaces continue to thrive and achieve the best it can. We started GroupSpaces with the ambition to build a global internet-scale company, and it's in this light that it's now time for me to move on to tackle the next challenge. However I'll continue to be involved in supporting the business, while my departure from the day-to-day has made room for fresh blood to drive it forward in new ways. I look forward to seeing GroupSpaces continue to grow, either standalone or in time with an appropriate partner.

With the bittersweet benefit of hindsight, I've been able to dwell on the key mistakes we made and how they affected our progress. For the possible benefit of others - along with bringing no small amount of closure for myself - I've detailed my thoughts on these mistakes and learnings as follows:

Failing to appreciate a lack of product-market fit as we moved into new markets

GroupSpaces originally started out serving the student market, where we achieved considerable success. We designed a tool to solve our own problems, discovered that there were many like us, and with the benefit of the closely-linked geographically-concentrated nature of universities and colleges we spread often like wildfire though campuses in the UK. A stand-out memory was signing up a single organisation at the London School of Economics at the beginning of the semester; with no further marketing we had 100 organisations using the platform by the end of that term.

The problems came when we followed this by focusing on marketing and distribution when expanding into other markets beyond students, failing to identify early on our issues with significantly different activation rates due to issues such as far less recognition, peer endorsement and good old-fashioned offline word of mouth. As a result we ploughed on for too long "pushing uphill" without going back and ensuring our product was fully appropriate for these new markets.

Insufficient focus, focus, focus

I firmly believe startups should choose their One True Thing as early as possible and aim to do that one thing well. With a small team that peaked around 12 people, we simultaneously tried to work on the product, while scale up user acquisition and paid marketing channels, while building out a business model consisting of 3(!) different revenue streams - all while attempting to fit our achievements into a framework that would enable us to demonstrate progress and hit

milestones that we could use to raise further funding. This had the inevitable effect that we ended up scoring straight Cs - sub-standard product, sub-standard metrics, sub-standard traction.

The irony is that at the time we believed we understood this point, and considered ourselves to be very focused - we'd learned to say "no" to most opportunities and suggestions, and had a very clear idea of our target customer, our product and what we were not. Our problem was that our "one thing" had many multiple parts. I cringe when I recall our spreadsheet models that had customer lifetime value as a messy fudge blending multiple different metrics, each of which depended on other metrics - and ultimately too many assumptions to meaningfully test.

I believe a startup needs to achieve a single A grade in one thing to advance to the next stage - if I could go back, the advice I would give myself would be to force ourselves to limit our focus to one thing - a single KPI at a time - to which all our efforts should directly and measurably contribute.

A confused strategy, attempting to hedge ourselves

Following on the point about focus, looking back we were very confused about which milestones we needed to hit in order to be able to progress to the next stage and raise subsequent funding. We ended up hedging ourselves - attempting to limit the downside in the case the upside didn't work out, attempting to have multiple stories and metrics on which we could raise if we didn't hit other targets. Predictably, there's no such thing as a free lunch - we simply ended up in no-mans' land with our straight C grades - no massively scaling user base on the social side and mediocre numbers on the business model.

We fell into this trap in part though the history of the company: we spent the first years of the company building a smaller business with mentors and investors focused on safety and predictability of building a solid company. Our ambitions led us to raise a VC round where the game becomes the go-big-or-go-home play (aka. throw-stuff-against-the-wall-and-see-what-sticks..). Not to say that either would have been a better or worse path to choose, but our mistake was not to make the jump to a boom-or-bust game fully, and instead get stuck in in the middle - a murky territory without clear strategy where every decision feels at odds with another, and our hedging to limit the downside definitely held us back from achieving the upside.

Premature scaling

It follows on from the points above, but we most definitely committed the all-too-common sin of premature scaling. Driven by the desire to hit significant numbers to prove the road for future fundraising and encouraged by our great initial traction in the student market, we embarked on significant work developing paid marketing channels and distribution channels that we could use to demonstrate scalable customer acquisition. This all fell flat due to our lack of product/market fit in the new markets, distracted significantly from product work to fix the fit (double fail) and cost a whole bunch of our runway.

Moving too slowly after we raised our Series A

(Back in 2010 when we closed the round, $1.3m was still known as Series A - there was none of that seedy talk you get nowadays..)

In hindsight it's painful that we were not up to speed with the full team we wanted to hire until a full 8 months after we closed our VC round. We proudly described how we'd spent 3 months "learning how to hire" - by doing everything we could to learn from others, interview carefully and ensure we hired the right people. "Hire slow, fire fast" was the advice on many a respected startup blog. Well, not to add to the problem - but as I now know: "hire fast, fire fast" is for sure the better plan.

Nothing struck me as such a comparison as when a friend more recently made a hire for her startup on the spot after an interview that went well, extended to 90 minutes and then - done. When I mentioned this to her she said simply "I need to hire 20 people in the next month". Granted not how it goes in every situation, but another piece of advice I would give my earlier self would be to trust my gut sooner without dragging out an interview process to multiple rounds as standard or because "regardless, we should spend some time making this hire to see who else comes along".

We should have closed our round being able to name the 3 people we wanted to hire next, and being moving forward producing product and traction with a full team (not necessarily large, but balanced) within 2 months. At the time it was

easy to feel like all our time on the meta-aspects of building a company was well spent. But for too long we were not 100% focused on the only things that ultimately mattered - customers, product/market fit, traction - and all the while our monthly burn was - well - burning.

What we did right

Despite all the above, we certainly did a number of things right. You may note that nowhere have I mentioned the key ideas behind GroupSpaces as among the things I consider mistaken. With GroupSpaces we have provided a great product that provides significant value to our many loyal users, and more continue to sign up as customers each day. I still believe a place exists for GroupSpaces or a similar service to scale to serve the market of the millions of membership organisations that currently rely on the unhappy marriage of a Yahoo/Google Group coupled with an Excel spreadsheet/google doc for member records, in addition to their own website and other services for events or payments.

I hope to write more about my learnings as time allows, but for now it just remains for me to restate my heartfelt thanks to those who have supported us in the journey with GroupSpaces, and look forward to what the future holds for both GroupSpaces and Stripe.

Original article here: http://insomanic.me.uk/post/48136679276/looking-back-at-7-years-with-my-startup-groupspaces

Startup Failure: Zillionears.com

My Startup Failed. Fuck.

I finally said it, my startup failed. Fuck. I felt like I was coming out of the closet when I first stated it aloud to my co-founder. We both knew for months it was not working out, but we never explicitly defined our situation as a failed one. Now that the elephant in the room has a name, we'll call him "Dumbo" which stands for "Didn't Understand Markets Brain Outline". That right there was our main problem. Our market demographic was musicians, and although a few of us had worked around the industry, we concluded recently we were not music SALES domain experts.

The product was a flash sale platform for musicians to release their music using dynamic pricing (zillionears.com). To us, this software was a no brainer for musicians to use. The artists get to engage their fans while enticing their community to share with friends. So we talked to a few artists who said they thought it was a cool idea. BOOM! Our idea had been validated! After that moment we basically stopped talking to artists for a year and built (and rebuilt) the software until we thought it was acceptable.

Our first beta test was a disaster when Amazon (who was our payment processor) suspended our account for not complying with money transfer issues. Fans were able to participate in the sale, but we were unable to capture their billing. We ended up paying the artist out of our own pocket and giving everyone his music for free (and we never told him that happened until now).

From that beta test we found out that our software needed to be rewritten to comply with Amazons terms. More importantly though, people really didn't really LIKE anything about our product. No one that used the service thought it was that cool. In fact, some people that participated in the sale didn't even like our "dynamic pricing" system. They were trying to support the artist, so saving a few dollars didn't excite them. They could easily have just gotten his music for free elsewhere.

We should have packed it up early right then, but we felt like we had already gone too far to quit. We rebuilt (and re-designed) the majority of the software, got approved by Amazon, and reached out to over 1,700 artists (each individually through different platforms). We got between 1 and 10 artists interested. Again, this just screams "PUT IT OUT OF ITS MISERY!" But we kept going. Finally the day came for our second beta (which was totally gonna kick ass for sure). The artist we had on board set up his sale page and was ready to go. Only problem is he totally misunderstood what our software was all about. Once he found out about the dynamic pricing he tells us "I think I am just going to release with another platform." FUCK! Are you serious????

After that we spent another month slowly letting it linger in our day to day lives. We went for one last ditch effort to make a press release, but couldn't get a single artist (out of the 1,700+ we talked to) to run a sale. My co-founder called me to tell me this news. I asked him "Would you like to use my gun?" I was referring to the scene in The Social Network where Zuckerberg's lawyer asks Saverin "Would you like to use my pen?" to manipulatively sign his shares over. I, of course, was referring to shooting this fucking company in the head and moving on with our lives! He agreed. We took Zillionears out back, and shot it in the head. It felt good.

Although our company did not succeed the way we would have hoped for, we all learned more in the past year than we had in college. Our insights and experiences have been invaluable. For each of my future posts I will go into detail about the things I learned while on this journey, and how to apply the knowledge to future startups so you can avoid ending up in a room with "Dumbo"!

Original article here: http://nemrow.tumblr.com/post/47728450959/my-startup-failed-fuck

Startup Failure: Everpix

Out of the picture: why the world's best photo startup is going out of business

One day last month, the seven employees of Everpix gathered at their co-working space in San Francisco to discuss the company's impending shutdown. Wayne Fan, one of the co-founders, opened a mock-up of the screen that the photo storage service's customers would see once the company announced the news. The screen described the refunds that would be offered to the company's 6,800 paid subscribers, assuming Everpix could come up with the money. No one knew if they would.

The immediate concern in the room was a forthcoming bill from Amazon Web Services, which hosts the 400 million photos stored with Everpix; the team estimated the bill would be about $35,000. "Our AWS bill is going to be due on the third. We're not going to be able to pay," said Pierre-Olivier Latour, who had the idea for Everpix four years ago after a vacation left him struggling to organize the hundreds of photos he took on the trip. Behind him, a poster advertised San Francisco's minimum wage of $10.55 an hour, which he had been paying his employees for the past month. "Amazon is going to reach out to us saying, 'Your card doesn't work.'" He paused. "So that's going to be fun."

In two short years, Everpix has gone from a dream shared by two French graphics experts to one of the world's best solutions for managing a large library of photos. It attracted 55,000 users and earned enough each month to cover the cost of the service, if not employees' salaries.

But while its talented team obsessed over the look and features of its product, user growth failed to keep pace. Starting in June, Latour tried to raise $5 million to give Everpix more time to become profitable. When those efforts faltered, he began pursuing an acquisition. Everpix had tentatively agreed last month to be acquired by Path, according to a source close to the social network. But Path's executive team killed the deal at the last minute, leaving Everpix adrift.

With the service's remaining lifespan now measured in days, Latour scrambled to find a home for the product and his team. He worked on one potential deal for a so-called acqui-hire, in which the team would stay together at a new company while the product shut down; and another for a true acquisition, in which Everpix's core technology would live on in some form. He also tried to negotiate a last-minute, product-saving infusion of cash from an investor who loved the product.

The question in the room was whether any of it would come in time. The bills were starting to come due, and there was no money to pay them. Silicon Valley prides itself on being a place where great ideas win, and yet here was an excellent product teetering on the edge of oblivion. "It feels like we're going 100mph into the brick wall," Fan said. "And we're still picking up speed. But we don't know if the wall is going to be there."

Picture this

Everpix began as the brainchild of Latour, a 34-year-old French entrepreneur who sold his first company to Apple in 2003. That company, PixelShox Studio, made motion-graphics software that was later renamed Quartz Composer. It became a key part of OS X, powering the operating system's screen savers and iTunes visualizations, among other applications.

At Apple Latour met Kevin Quennesson, a fellow French computer scientist who joined Apple in 2006 and became the project lead for Quartz Composer. Quennesson's background is in math and physics; his name is listed on six patents related to motion graphics, image processing, and graphical user interfaces.

"The more photos he took, the less likely he was ever to look at any one of them ever again." By 2009 both men had left Apple and were working at Cooliris, which makes photo viewing software. Latour set up an office for Cooliris in Japan, and after spending some time traveling through Asia with his girlfriend, he became frustrated with how difficult it was to store and organize all the photos he was taking. He discussed an early idea for a product with Quennesson, who was interested in using math and science to create better photo software.

Quennesson had noted that the more photos he took, the less likely he was ever to look at any one of them ever again. "People take more and more photos, but paradoxically, they become more and more disconnected from them," he said

last month in a conference room at their co-working space. "You don't want to go back to this whole life that you've captured, which is counterintuitive. It's the most important thing — it's your life! So there's this obvious problem."

In August 2011, Latour incorporated under the name 33cube Inc. so they could refine his prototypes. In June, after searching for designers on the portfolio site Dribbble, they came across Wayne Fan, who worked at a San Francisco firm doing interaction and visual design. Fan, too, felt frustrated by photo software, and they brought him on as a co-founder. "We give consumers these tools like iPhoto or Aperture or Lightroom, which are basically artifacts of this bygone era," Fan said. "Why would anyone need to rate a vacation photo from one to five stars, or flag them? Or have a histogram with levels? That's crazy."

To give the team a deadline, Latour entered the startup competition at TechCrunch Disrupt. They spent the next few months building a prototype of their service, which they were now calling Everpix. Many of its most distinctive features were in place from the start. The service seamlessly found and uploaded photos from your desktop and from online services, then organized them using algorithms to highlight the best ones.

The software was fast, the design was clean, and the service was simple to use. "The best part about Everpix may be its 'set it and forget it' nature," TechCrunch noted at the time. "After the one-time installation and configuration, there's nothing else you have to do." To the team's surprise, Everpix became a finalist at the competition. (They lost the $50,000 first prize to Shaker, a bizarre kind of Second Life-meets-Facebook social network that raised $15 million and hasn't been heard from in a year.)

Everpix had gotten started with small investments from former Apple executive Bertrand Serlet and 500 Startups, the incubator run by former PayPal executive Dave McClure. In the wake of Disrupt it raised money from investors including Index Ventures, Strive Capital, and Picasa co-founder Michael Herf, for a total of $1.8 million. "The use case was incredibly compelling," said Nuno Gonçalves Pedro, managing director at Strive. "The product really looked very robust. And finally, we really liked the team."

Everpix had acquisition discussions after Disrupt with Facebook and Dropbox, among others, but waved them all away: the founders wanted to create this service on their own terms. They spent the next six months building their product in a public beta before releasing what they considered to be the true Everpix 1.0 this March. A free option let you see all your photos from the past year, or longer if you connected your desktop computer to the Everpix iOS app. For $4.99 a month or $49 a year, the service would let you store an infinite amount of photos. A feature called Flashbacks sent users daily emails of their photos from the same day in prior years, which led to a huge spike in the number of users who returned to Everpix daily. Critics raved. But it had taken the company longer to get there than investors would have liked. "At that point, it had been one and a half years since we started," Latour said. "Which is a huge amount of time for a startup to have a full-featured product."

While the team obsessed about perfecting the service, the founders paid less attention to the subject investors care about most: growth. Everpix built some features for sharing photos, but there was little else in the product to help it spread to other people. At one point, the team considered requiring a user's friends to create an account to download any photos that the user shared with them. It was a surefire way to boost signups — but also felt like the sort of ugly, needy design choice that the team prided itself on avoiding. The idea died.

And so at a time when successful photo apps were attracting users by the millions, by March the company had attracted fewer than 19,000 signups. Everpix had spent almost nothing to advertise the service, not that it could have afforded to pay for much. The company had spent almost all of its $1.8 million building the service.

As May approached, Everpix was going broke.

Into the crunch

In one of the last breaks Everpix would ever get, Index Ventures agreed to loan the company $500,000 in the hopes that it would float the team until they could complete a proper fundraising round. (500 Startups contributed another $50,000 as well.) Latour hired a marketing specialist, Julie Supan, who previously had helped YouTube and Dropbox position themselves in the market. Together they came up with a tagline for the service: "solving the photo mess." With a re-written pitch deck on his laptop, Latour set about trying to raise money.

Given his connections at Apple and the investors he had already attracted, Latour had no trouble persuading Silicon

Valley's top venture capital firms to take a meeting with him. His goal: a series A round, in which Everpix would trade equity in the company for $5 million in cash. The money would support the team as they worked to build their subscription business and become profitable.

Unfortunately for Everpix, they went out to raise money in the midst of what has become known as "the series A crunch." The number of initial (or "seed") investments has increased dramatically in the past few years, while series A investments have plateaued. Many investors remain willing to write a $100,000 check to see if a startup becomes an overnight success. But when it comes time to write a $1 million check, or a $5 million check, they have become much more selective.

The sheer explosion of apps, and the ease with which consumers can switch between them, has spooked the investment community. "Investors have stepped back a little bit," says Gonçalves Pedro, of Strive Capital. "They see this as a dramatic increase in risk."

In meetings on Sand Hill Road, Latour says, nearly everyone expressed enthusiasm for Everpix's product. But one by one, they turned him down. After two meetings with one well-known firm, a partner sent Latour an email. "You guys seem to be a spectacularly talented team and some informal reference checking confirmed that, but everyone here is hung up on the concern over being able to build a >$100M revenue subscription business in photos in this age of free photo tools." Said a partner at another firm: "The reaction was positive for you as a team but weak in terms of whether a $B business could be built."

Meanwhile Index, which had led the seed investment and provided the $500,000 loan, declined to lead the series A round, signaling a loss of confidence in the company that surely resonated among other investors. "While the product was clearly superb and had a very small but very loyal following, we were not comfortable enough with the other aspects of the business to kick up our level of investment," said Neil Rimer, a partner at Index.

With fundraising efforts hitting a dead end, Latour turned to the idea of an acquisition. Non-disclosure agreements prevent him from talking about the discussions. But the tech giants that once sought Everpix were no longer interested, for a variety of reasons. And when Everpix finally reached a tentative deal with Path last month, it collapsed before the final paperwork could be signed. The social network, which has had fundraising problems of its own, laid off 20 percent of its staff a few days later. Path declined to comment.

Latour continued to seek new investments and to court potential acquirers. In the meantime, his employees set about the grim work of dismantling their product. They agreed to work for minimum wage in the hopes some deal would be struck. "I haven't given up on the product yet," said George Leontiev, the company's 25-year-old iOS developer, last month. "Maybe someone will come along and see that and help us do what we want to do."

But as time ran out, hopes diminished. "It succeeded in every possible way," said Jason Eberle, who built the web version of Everpix, "except for the only way that matters."

Lessons learned

The founders acknowledge they made mistakes along the way. They spent too much time on the product and not enough time on growth and distribution. The first pitch deck they put together for investors was mediocre. They began marketing too late. They failed to effectively position themselves against giants like Apple and Google, who offer fairly robust — and mostly free — Everpix alternatives. And while the product wasn't particularly difficult to use, it did have a learning curve and required a commitment to entrust an unknown startup with your life's memories — a hard sell that Everpix never got around to making much easier.

Rimer put it a bit differently: "Having a great product is not the only thing that ultimately makes a company successful."

On the other hand, they created a product that people genuinely love. The iOS app has a 4.5 star average rating, out of more than 1,000 reviews. About half of free users return every week, and more than 60 percent return each month. And 12.4 percent of free users wound up subscribing to the service, compared with a rate of about 6 percent for Evernote, which also sells its service on a freemium basis.

He wanted to let the users of Everpix know: "We tried." Back at the Everpix office, the team focused on how to shut down in the gentlest way possible. They hoped that at the very least, they would be able to refund their customers' money and

allow them to download their data. "We've made a commitment to people," said Sameer Sundresh, the company's back-end engineer, wearing a pained expression on his face.

Meanwhile, someone had to write the shutdown letter. The task fell to Fan. He was struggling with the words, but knew what he hoped to get across. He wanted to let the users of Everpix know: "We tried."

The end

In the final hours of the company's life, Latour's efforts to keep the team together unraveled. The acqui-hire fell apart when the potential acqui-hirer decided it didn't want the entire team. The investor who loved the service made a tentative offer to float the company for another month, but Everpix's employees worried that would only prolong the inevitable. Meanwhile, Index and Strive pushed back against the offer, as it reduced their stakes in the company to nearly nothing. And so Everpix is in the process of selling off its technology in order to cover the costs of ending the service. The team estimates it will cost roughly $200,000 to refund Everpix's customers, allow them to download their data, and pay employees their full salaries. None of users' photos or personal data will be transferred as part of any sale.

On Thursday, the team packed up their office in SoMa. Today, they will post their shutdown notice from borrowed office space elsewhere in the city. One or two employees will continue to maintain the service through the end of the month.

In time, Latour hopes, the lessons of Everpix will become more clear to him. Where had it all gone wrong, exactly? Maybe there was something obvious that everyone had missed. Maybe it was ahead of its time. Maybe not. In any case, Latour said, he would be fine. "You look at all the problems that we've had, and it's still nothing," he said. "I have more respect for someone who starts a restaurant and puts their life savings into it than what I've done. We're still lucky. We're in an environment that has a pretty good safety net, in Silicon Valley."

After they packed up their office, the team went out for lunch. Over beers, they discussed a man who had tweeted at Everpix that day asking them to add some navigational arrows to a section of the web app. "How hard can it be?" the man asked. How hard can it be? Latour read the tweet with deep amusement. Everpix itself had once been that naïve. But not anymore. Latour stood up to take a picture of his team. In the photo, everyone is laughing.

Original article here: http://www.theverge.com/2013/11/5/5039216/everpix-life-and-death-inside-the-worlds-best-photo-startup

Startup Failure: HelloParking

HelloParking postmortem: a look back, and a new perspective

More than two years ago, Neil and I shut down HelloParking. We never managed to find product-market fit, and we were having trouble scaling. Before heading back to class and moving on, I wrote about our experience. In a rather long-winded essay, I exposed our triumphs, our failures, and told the story of HelloParking, a failed parking startup, to anyone who felt compelled to read it.

To my surprise, those who felt compelled numbered in the tens of thousands, and I've been contacted by many ever since publishing our story. As it turns out, we weren't the only ones wanted to disrupt the parking industry. And as it turns out, where I thought we went wrong two years ago isn't the full story.

After writing HelloParking's postmortem, I've since moved on – I've learned about Lean Startups and Customer Development. I've ripened my understanding of how to shape products. And as all things on the internet have a tendency to do, my original post has lived on. Because this one morsel of my past life continues to attract readership, I've decided it's time to share an epilogue.

New insights: To save the average reader from the toils of reading through the pages that follow, I've compiled my summary of new insights in the bullets below:

In the University of Entrepreneurship, Customer Development is a pre-requisite, and there's no such thing as graduation. What you think you know is probably wrong. You can't getting inside your customer's head without getting out of the building. Any attitude otherwise is doomed to fail.

It's not that the parking industry wasn't ready for us. We weren't ready, and didn't know how to speak their language.

Pivoting for pivoting's sake is worthless. It should be a calculated affair, where changes to the business model are made, hypotheses are tested, and results are measured. Otherwise, you can't learn anything.

When it comes to entrepreneurship education, nothing beats a good reading list and personal experience.

Having graduated with a degree in Entrepreneurship, I thought I was ready to fire on all cylinders. I was wrong. Nothing has prepared me more for success in the startup world than a handful (a roomful, rather) of good books. Works by Steve Blank (Customer Development Guru), Eric Ries (Lean Startup Guru), and the Heath Brothers (Applied Psychology Gurus) are great places to start.

Ignore the hype. Focus on product-market fit, earlyvangelists, and bright spots. Along our journey, lots of really cool stuff happened. We got featured in the press (press, press, press, and more press). We were finalists in business showcases. We (nearly) made it to TechStars. I confused this excitement with success, and I should have seen it as a distraction.

Broadcast knowledge, and share failure. It opens doors and is helpful to others. Failure doesn't need to be guarded, and shouldn't be a source of embarrassment.

Writing about what we learned attracted new connections and got me recruited into my current position, where I have been for two years.

"Why do you think you failed?" is a question I continue to get asked.

Years have passed, and with them I've grown a deeper understanding to the answer here. Our first model ("AirBnb for parking spaces"), has since sprouted dozens of fellow upstart imitations, without much success. Just like ours, many who started in this model have since graduated to the commercial parking space.

It seems that the reason the AirBnb model hasn't quite worked yet is the difficulty in collecting enough supply. Some have found marginal success at the hyper-local level, but none have found a model that works at scale.

More folks are hunting for parking spaces than exist parking space owners who are willing to share. This is the problem we weren't able to solve, and a problem that still exists today for folks taking their own stab at the driveway sharing

business. That's not to say it's a problem that can't be solved, but rather it's a tough one that has yet to be solved at scale (that I've seen, anyway).

And then we pivoted. And pivoted. And pivoted…

Our failed attempts at the "ZipCar for parking", "Priceline for Parking" and "Groupon for parking" models followed in the months thereafter. And then we gave up. At the time, I was convinced that the industry wasn't ready for our innovations. I was convinced that property managers and parking operators didn't "get it" or just didn't care.

That just can't be the case. And it wasn't.

I'm now the product manager at QuickPay, where I have been for two years. I know from first hand experience that parking operators are willing to innovate. They're looking for ways to grow their businesses. But at the time, I just didn't know how to speak their language. I didn't know a lot of things.

I did everything wrong.

First of all, I didn't have a good understanding of the Customer Development model. And though I didn't realize it at the time, doing "xyz for parking" isn't the way to build product and discover value. The transitive property doesn't apply as well in the startup world. ("If x worked for industry y, then x must work for industry z")

The truth is, as co-founders of HelloParking we huddled together to decide on ideas that sounded nice, built prototypes, put on our salesman hats, and didn't understand why we weren't closing deals. And then we rinsed and repeated. We pivoted because pivoting seemed like the right thing to do.

But we never defined clear hypotheses, developed experiments, and we rarely had meaningful conversations with our target end-users. And while we had some wonderful advisors in the parking industry, we should have met with everyone we could get our hands on. Worst, we rarely got out of the building.

But at the end of the day, I'm fortunate to have had the opportunity to learn from my mistakes in my early twenties. And I'm fortunate to have been given a second chance as an early employee at QuickPay. I'm better for the experience, and have learned from it. I only hope that some of you out there can learn from my mistakes as well.

Original article here: http://chrishoog.com/the-helloparking-postmortem-a-look-back-and-a-new-perspective/

Startup Failure: Gowalla

Play by your own rules.

A little over a year ago I joined Facebook as a Product Manager. I work with the Pages, Location, and Events teams. As I've spent a decade of being the founder of three different companies, this has been a year of change for me. Along the way I've piled up a supply of stories (and lessons learned) I somehow feel compelled to document—but the reality is I'm crap at journaling. I just need to start when the story is fresh and close at hand.

Today that means it's easiest to talk about Gowalla, a startup at least a handful of you are familiar with. We built a few amazing things. We also had our fair share of disappointments. Of course, you might surmise that our disappointments ended up outnumbering our amazing things. I won't argue with that. It's part of the game we play in Startup Land, fighting against the gravity that's always working to pull our ideas brutally back to the ground.

I would like to believe that these stories aren't just applicable to startup life, but to product creation in general. But since I'm no longer running my own company, I've gotta believe that, right?

So here goes…

You fool! You fell victim to one of the classic blunders—the most famous of which is never get involved in a land war in Asia—but only slightly less well-known is this: Never go against a Sicilian when death is on the line!—Vizzini In January of 2008 some friends and I launched PackRat, an early social game on Facebook. Eight months later, in August, we gave PackRat a cash-backed virtual currency. By December we were turning over $200,000 in monthly revenue. Not too shabby for a team of eight.

About the same time we began exploring themes for other game-like experiences to build. A few ideas were basically PackRat re-skinned for a different audience. Another was a quasi-real-time strategy that would reward you for polluting the world (I kid you not, it was going to be awesome). But only one of our ideas truly stood out as being unique.

The first iPhone with GPS had just landed, so we were enthralled with the idea of mixing art, game-like incentives, and real-world locations to inspire people to go explore and share their favorite places and experiences. Eventually we would refine this goal as a way to see the world through the eyes of your friends. We believed this idea to be the craziest of the lot, so we set about on a rigid pace to launch it in time for SXSW (these were the the days you still had a chance of being heard).

Of course, this idea was Gowalla.

A week before launch I came across some interview with Dennis Crowley in which he described a new project that he and Naveen Selvadurai had been working on. It was called Foursquare.

A week later Gowalla and Foursquare would launch on the same day.

I remember sitting in Halcyon on 4th St. on the eve of SXSW as a few of us downloaded Foursquare to give it a spin. While it was, at the time, quite different in execution from Gowalla, the common presence of fresh terms like Check-ins, Pins, and Badges would be the first sign to the public that a new wave of location-based services had arrived.

At launch, both apps had their distinct moments of strength and weakness. We thought Foursquare was crap, and believed the design nerds flocking to Gowalla validated our attitude. Gowalla also worked anywhere—we were the first to crowd-source a local database from scratch. Foursquare only worked in a dozen cities. In short, all else equal, we believed people would use our service because of its superior craft and availability.

Of course all was not equal. Foursquare was not Dens' first rodeo (even prior to Dodgeball, Dens' involvement with earlier location-based projects like Vindigo continues to impress me). Foursquare's first cut proved more accessible and less fussy to the common crowd than our own. And while Foursquare was initially only available in a dozen cities, a dense and deeply connected crowd in New York—many of them previous Dodgeball users—had already formed a strong base to build from. The New York media, thrilled that "the next Twitter" might come out of Manhattan, also proved to be a force.

Since time to fix our flaws was in limited supply, the next six months turned into a footrace, one that would carve a rut difficult for us to get out of philosophically. We entered a check-ins arms race.

We significantly backed off development of PackRat and bled its profits in order to speed up our pace.

Foursquare would soon raise a round of funding from some of the brightest investors in the game. They began to publicize the number of people who were checked in every day, and consequently defined the metric that would be used to measure success in our space. In doing this, they created the game. Since we were the other major app at the time with a BFB labeled Check In the pressure was on to return in kind.

On the heels of their financing we would launch a number of new social features along with support for Facebook Connect. Our growth took off. We started to close the gap a bit with regard to daily check-ins.

Investors began to call. So did a small handful of companies who expressed interest in acquiring what we had built.

In October of 2009 I flew from Austin to San Francisco. I remember calling Rachel after a few days and telling her that I wouldn't be home for a bit. I ended up staying two weeks. I met with eighteen different venture firms, telling each our plan to scale (a post for another day). I left San Francisco with a small stack of term sheets, and soon we agreed to raise $8 million and change from a first-class group of investors.

Money in the bank, we were ready to go to war. The hype was on. The network effects were at work.

The growing public attention sparked by the investment became both a heaven and a hell. On one hand, during the course of Gowalla's life we received more tech press coverage than is right and holy for any startup. On the other, I dare you to find a post about Gowalla that doesn't mention Foursquare.

The Check-in Wars had begun and everybody knew it. The tech world wanted it some drama. We were happy to oblige.

With SXSW right around the corner again, all the discussion centered on the perceived brawl that would go down in Austin. The "winner" would be measured by check-ins.

We were the hometown underdogs. We had to show up. And we did. Both companies had sponsors. Both had swag. Both had parties.

(Our party was better. Just sayin'…)

We put up incredible numbers, far beyond anything we had ever done before. We received a number of awards and honors related to the festival (Scott, if you're reading this, it's my turn to have the trophy back for a year). In short, we felt on top of the world.

SXSW set the rivalry in stone. Wired UK even memorialized the happenings in The Great Check-In Battle. Everyone knew the score, and how it was being kept. And for as well as we had done during SXSW, when the dust settled, outside of Austin we were still number two.

A couple months later, Foursquare would raise $20 million from another crew of awesome investors, valuing their company well over $100 million. If you were to come up with a "check-in to post-money" ratio, the pricing of our approximate $8 million raise and their $20 million raise were right in line.

Unfortunately, once your key metric is tied to cash value in the eyes of investors, it sucks to be number two. Your ceiling has been bolted in place. Your future capacity to raise cash or sell has a lid on it now.

We felt that in order to survive we had to get our numbers up. We tried just about everything to juice growth, some ideas being more successful than others.

Foursquare did a marketing deal with New York Times. We did one with Washington Post and USA Today.

Foursquare built an Android client. We built an Android client. Foursquare had a Blackberry client. We needed a Blackberry client. Heaven forbid we lose those Blackberry users!

In one of the many highlights of Gowalla, we crafted an amazing tie-in with Disney that was loved by both our community and theirs. Each of these endeavors cost us time and money. Unfortunately the relative payoff for us was simply less due to network effect.

While Gowalla continued to grow, the trajectory was not what it needed to be. At least not in terms of winning the game we had chosen to play.

We were the younger, prettier, but less popular sister of Foursquare. And even that had changed. In time, Foursquare had dramatically improved the design and experience of its service. This was no longer a defensible platform for us as a company.

Around this time we knew that our path was in trouble. We would have to pull out the stops to change our game.

The months that followed brought new product iterations from us. We gave people a way to check-in on both Facebook and Foursquare through Gowalla. It was very well received. But it was a lot like being Tweetdeck instead of Twitter. Tweetdeck might be cool, but let's be honest, you'd rather be Twitter. We were still in the game of brokering check-ins, a game we couldn't win.

That said, I'm ridiculously proud of the product our team built during this time. It was revolutionary. We were Editor's Choice in the iTunes Store. Featured by Google Play. Even the webOS folks at Palm loved us! Hardly a week passes today that someone doesn't tell me how much they loved and now miss Gowalla. We built something special, but had chosen to measure our success in a way that kept us tied down.

Our time was running out.

By the time SXSW rolled around again in 2011, the hype surrounding our rivalry had started to fade as the tech community had began the search for The Next Big Thing™, this time anointing Group Messaging as the new hill to be conquered in place of Check-Ins.

Truth be told, we didn't really care about check-ins either. It was just the action verb we put on the orange button that performed our app's primary function. What we really wanted was for people to see the world through the eyes of their friends.

Now here's the M. Night Shyamalan moment for you:

It turns out there was another app that shared a similar vision called Burbn. They were building yet another check-in type service loaded with every feature but the kitchen sink. But early user feedback, coupled with a desire to avoid the check-in battle shitshow already in progress, led them to drop everything to focus on one simple feature: photos.

They made the act of taking and sharing photos (many of which just happened to be location-tagged) fast, simple, and fun.

They made their own rules. They called it Instagram.

That whole see the world through the eyes of their friends thing? Turns out Instagram did a pretty good job of this.

While we were busy playing tug-of-war over check-ins, someone else found a path to the goal with less friction.

About a year after the launch of Instagram, Gowalla's service would shut down and several of us would join Facebook. Others would move on to new endeavors of their own. Ironically a couple from the team would join Instagram.

(I'm really happy I still get to work with those guys.)

Of course there's a lot more to the story of Gowalla. I hope to unpack other parts of the journey in time—growing a team, raising (and spending) money, what it was like to sell, etc. But of all the lessons I've learned through this journey, this one sticks out like a splinter.

Play by your own rules.

Listen to your users more than the press. Don't get sucked into the gravity hole between you and your competition. Ruthlessly run your own path, not someone else's.

Today I'm at Facebook. It's a special time to be at the company right now. We're able to build unique products that few others can dream of. But those same challenges still present themselves: How do we choose our own path? Build to our own strengths? Avoid the gravity holes?

These are the questions I chew on every morning.

Original article here: https://medium.com/work-education/play-by-your-own-rules-6152adc41de9

Startup Failure: Sonar

Postmortem of a Venture-backed Startup

For those unfamiliar, Sonar Media Inc. was a mobile app created to help make the world a friendlier place. Our mobile app buzzed in your pocket when friends were near and ushered in a new wave of "Ambient Social Networking" companies. Downloaded by millions of people all over the world, Sonar was promoted by Apple and Google in 100+ countries, won numerous awards such as runner-up at TechCrunch Disrupt and Ad:Tech Best Mobile Startup, raised nearly $2,000,000 from prominent angels and VCs, and was featured on more than 300 publications including the New York Times, CNN, CNBC, TechCrunch, and TIME.

And yet, we failed.

We did lots of things right and lots of things wrong at Sonar. Below I do my best to share a few of our lessons learned.

The Search For Product/Market Fit

"Make something people want." —Paul Graham Listening to your users: False positives We launched Sonar with Facebook, Twitter, and Foursquare support. Shortly thereafter, users buffeted us with requests for Linkedin integration. Ostensibly, they wanted to use the app to meet fellow professionals.

Eager to please, we rushed to add Linkedin. The net effect? Nada. My guess is that the people asking were not actual users, but rather people that "wanted to be" users. We had mistaken noise for signal.

Lesson learned:

"I would use your product if only you had X feature" is a dangerous signal to follow. Users do their best to anticipate what they want before they've seen it but, like entrepreneurs, they are often wrong.

Enterprise companies should validate demand by asking customers to put their money where their mouths are. Media and social networking companies should double down on analytics to find, observe, and build for actual user behavior.

Listening to your users: False negatives One of the most requested features was a "map like foursquare" for our check-ins. Instead, we appended a simple "@Sonar" to content that users shared from our app. Although we had designs for a map, we never got around to building one. We were too busy building the future of ambient social networking!

Mistake. People didn't like the bland "@Sonar" text string so they stopped sharing updates from Sonar. Their friends never engaged with our updates in the first place. Facebook noticed this and started hiding our posts. Instead of optimizing for actual user behavior, we spent countless whiteboarding sessions trying in vain to design an alternative.

Lesson learned:

You are probably not the Steve Jobs of __.

Removing friction from existing user behaviors (e.g. checkins) almost always has a higher ROI than building castles in the sky (e.g. hypothesizing about your API). Find all the dead ends/local maxima in your current products before building new ones!

Growth vs. Engagement

We received conflicting advice from lots of smart people about which is more important. We focused on engagement, which we improved by orders of magnitude. No one cared.

Lesson learned:

Growth is the only thing that matters if you are building a social network. Period. Engagement is great but you aren't even

going to get the meeting unless your top-line numbers reach a certain threshold (which is different for seed vs. series A vs. selling advertising).

Things I Wish I Spent Less Time On

"Focus is saying no to 1,000 good ideas."—Steve Jobs Events

I realized the error of my customer acquisition strategy as I awkwardly made my way through a small Meetup I had just pitched. It was 11pm on a Tuesday, I was exhausted and still had real work to do once I got home. Yet there I was, in a shitty bar trying not to skewer anyone with my Sonar sign as I dodged person after person asking me to install THEIR app.

Lesson Learned:

Events are for research, business development, and hiring; NOT for getting to 10,000,000 downloads.

Brands & Agencies

When MTV, Kraft, Digitas, and the like reached out to us we weren't sure what they wanted. It took us at least 10 meetings to realize that, rather than delivering us millions of their customers on a silver platter, they were keeping tabs on us so that they could get access to OUR audience if we ever took off!

Lesson Learned:

Be polite, but postpone brand and agency "intros" until you've built your own audience. If you build it, they will come (and pay).

Corollary: Investors know this. You sound stupid when you talk about your impending "big deal" with "XYZ brand" that's going to drive massive customer acquisition and revenue.

Side projects

In the winter of 2011, we signed a partnership w/ Wired magazine to demonstrate our technology by providing visitors of their Times Square popup store with personalized in-store product recommendations.

That "small side project" cost us 6 weeks of development and delivered no appreciable benefit other than getting to hang out with the cool people at Wired.

Lesson Learned:

You do not have 20% time. Identify your top three priorities. Throw away numbers two and three.

Competition

In the run up to SXSW 2012 when the insider media had fabricated Highlight as heir to the throne and some of our more fair weather investors had written us off, my confidence was against the ropes. We reordered our roadmap to rush out comparable features but were now BEHIND. I put on my best brave face but inside my gut was rotting away. I still remember thinking on the flight to Austin "fck, we had it, and now we are going to lose it."

Oops! Highlight never went anywhere but we definitely wasted a ton of energy and sleep "responding to the threat" when we should have been figuring out how to make our own business work.

Lesson Learned:

Be steady at the wheel. The only way one startup can kill another startup is by getting into the other's head and leading them off a cliff.

If you don't believe me, try this proof. Are your competitors releasing a bunch of the same features that you have on your roadmap? Yes? Do you know what consumers want*? No? Great, then neither do your competitors. Get back to figuring out what users want!

*Hint: If you did, you would already have traction.

Selling the company

When the ambient social networking space iced over in the spring of 2012, Sonar's controlling investors decided it was time to "flip the asset." They connected us with a daily deals company looking for "Big Data" solutions. We stopped working on the app and devoted all of our resources to repacking our backend technology to solve BigCo's problems. Instead of paring down expenses to extend our dwindling runway, we piled on hires and ramped up our infrastructure.

The daily deals space imploded but we spent nearly nine months, dozens of meetings, and several hundred thousand dollars "selling" Sonar into a company that nearly went bankrupt.

Lesson Learned:

Companies don't get sold, they get bought. The best way to get bought is to build something of value. That's hard to do when you are trying to sell.

Misalignment

We built Sonar out of an incubator that I helped launch in 2010. To be absolutely clear, the incubator was instrumental to getting Sonar off the ground and helped us considerably along the way. Unfortunately, there are a number of structural issues facing incubators and the operators they employ. I address some of these below.

The decoupling of responsibility from control created ambiguity and confusion, tension and frustration for all parties. From day to day decisions such as negotiating an employment contract to company defining ones such as when to sell the firm, alignment was a constant challenge. Occasionally, we were simply at odds.

Perhaps the most detrimental aspect of the incubator model was not its potential for hinderance but its facility as a crutch. As someone responsible for building and running a company that I ultimately didn't control, it was far too easy to point a finger.

In my opinion, the most tragic example came when our incubator sat on a financing that would have rebooted the company. After nearly a month at loggerheads, our would-be investors gave us 48 hours to "take it or leave it." In hopes of saving the company, I made an ultimatum: we move forward together or I would have to walk away. No one budged, time elapsed, and our term sheet evaporated. I resigned as promised, blaming them for killing my baby.

Lesson Learned:

As John Burroughs said, "A man can fail many times, but he isn't a failure until he begins to blame somebody else." Avoid bad relationships like the plague but when you inevitably find yourself in a difficult partnership, don't waste precious energy wailing against it. Make it work or move on quickly.

It's All About People

"The essence of competitiveness is liberated when we make people believe that what they think and do is important – and then get out of their way while they do it."—Jack Welch

Be practical about team building

We lost our first would-be hire, a fantastic Google engineer. While we were debating his contract, he was taking a job elsewhere. Conversely, we hired another, much less proven, engineer on the spot. While he ultimately wasn't a great cultural fit and we definitely waited too long to part ways, he was instrumental in getting V1 out the door.

Lesson Learned:

If you are an experienced entrepreneur with lots of options, by all means, hold out. For most first time entrepreneurs, holding out risks never getting off the ground.

In the beginning, established people probably won't work with you. Prove yourself by finding diamonds in the rough, like yourself. With their help, you can level up your organization and convince the big fish to join.

Culture is your cofounder

We built an amazing team at Sonar. Everyone was extremely smart, passionate, dedicated, and hardworking. We celebrated milestones with tequila. We hung at the beach. Even when times were tough, everyone pushed as far as they possibly could, and then some. I have big love for all of my former colleagues and am confident they feel similarly.

That said, our culture was more of an emergent property than a deliberate choice. Sure, we had brainstorming sessions and posted goals prominently but most of our culture we absorbed from the people with whom we surrounded ourselves.

Lesson Learned:

Think of culture as a cofounder that is present when you are not. You are decisive, communicative, and respectful but it's your culture that helps everyone know how to act when you are out of the room. Give that voice clarity and authority.

The trick is to avoid hollow words. Since a startup's culture ultimately mirrors that of its founder, maybe the best thing that you can do is work hard to get clear on who you are. Write that down and share it with your team. If you've been honest, every action you take will reinforce your values.

Onward

"What we fight with is so small, and when we win, it makes us small. What we want is to be defeated, decisively, by successively greater things."—Rainer Maria Rilke via Tim O'Reilly

Startups don't die when they run out of money, they die when their founders let go. I ultimately stepped away from Sonar when I came to the conclusion that, despite all that we had invested, everyone stood a better chance starting anew. It's difficult to accept, but sometimes you have to concede a battle to win the war.

I am indebted to the hundreds of people that invested their sweat, money, love, and/or time into Sonar, be it three years of labor or a casual phone call. Special thanks to my amazing team, faithful advisors & investors, and supportive family & friends. Finally, huge shout out to the millions of users that gave Sonar a shot. Your stories about meeting your boyfriend on Sonar made it all worthwhile.

We started Sonar with a belief that everyone has the potential to be amazing and that technology can unlock that potential. My experience at Sonar has only strengthened my conviction. I can't wait to bring everything I've learned to bear on what's next.

Let's do this.

Original article here: https://medium.com/@brett1211/postmortem-of-a-venture-backed-startup-72c6f8bec7df

Startup Failure: Decide.com

Postmortem of a Venture-Backed, Acquired Startup

Decide.com was acquired by eBay on September 6, 2013. For those unfamiliar with Decide, we invented technology that predicted the future price of consumer goods. Thinking about buying a Samsung television? We'd tell you whether the price would drop in the next two weeks. We helped you decide when to buy.

I'm obviously happy with the outcome. From my brother's basement, we raised $16.5M, employed 30 people and most importantly built something people found useful. Only in my wildest dream would I have the audacity to think I'd build something that would be acquired by a company like eBay.

Although the outcome was positive, it's helpful to reflect on the experience. Below are 3 lessons I learned from the experience. They're not new but hopefully from within the context of a real startup they're more tangible.

Scale Your Company with Your Product

We launched Decide.com on June 20th, 2011 with 20 people and 2 round of funding totaling $8.5M. That's atypical, especially these days when investors want to see traction before investing. Instead of those resources helping us find market-fit it slowed us down.

We focused on company building:

Hiring. We hired 20 people in the 9 months following our first round of funding. That took lot of work. For every person you hire, you might interview 5 others, for each of those people you have a recap meeting with everyone that interviewed them, work on a offer for the person you want then sell them. That doesn't take into account the work it takes to source candidates and fire people that don't work out.

Organizational Structure. As we kept adding people, we had to figure out how they'd all work together. What teams should do what? Who leads those teams? How do those teams interact? Who needs to be in the meeting where we decide what we're doing?

Once we got through those parts, it was harder to change directions:

Communication Overhead. The larger the team the more energy you need to spend communicating. Let's say you want to change directions, something that happens often pre-market-fit. You have to convince everyone that there needs to be a new direction, what the new direction is and why this new direction will work when the last one didn't.

Specialization. As companies grow, the people they hire become more specialized. If we had pivoted to target businesses (we talked about it) instead of consumers we'd need sales people (we didn't have any) which probably meant we'd have to layoff other people (probably engineers) to hire them. This didn't happen to us but it impacted some of our decisions so I felt it was worth mentioning.

Lesson Learned:

Do whatever helps you reach market-fit faster. In my experience, that's raising modestly (if at all) and keeping the team small. 2 or 3 people can get a lot done and is small enough where you don't have to worry about structure or communication.

You wouldn't scale your technology to support 10M users when you only have 100. Don't scale your company that way either.

Hire Similar Minded People Our hiring criteria can be summarized with 2 basic questions: (1) Are they great at what they do? (2) Do we like them? Turns out there should've been a 3rd question. (3) Do you generally agree with me?

We spent a lot of time debating rather than doing. What technology should we use? When should engineering get involved with the design process? What's our design methodology? How long should our release cycles be? Should we

even have release cycles? I could go on indefinitely.

Lesson Learned:

Decide how you want do things then hire people that want to do things that way. There's value in having a diversity of opinion but in a early stage startups, the benefits (moving fast) of hiring people that generally agree with you outweigh the benefits (diversity of opinion) of hiring people that don't.

If you can't hire anyone that agrees with you, re-evaluate how you want to do things.

Expanding Your Target Market Doesn't Help You Find Market-Fit

We started with support for 3 categories (televisions, laptops and cameras) but expanded into all electronics within the first year. Looking for more growth, we talked about a few options:

Option 1: Expand into more categories. Be relevant to more purchases. Number 1 thing our users asked for.

Option 2: Expand beyond when to buy. Move up the buying funnel by helping people figure out what they should buy (product scores, reviews, comparison,etc.) in addition to when.

Option 3: Double-down on when to buy. Make it so good, if anyone is buying in a category we cover, they have to use us.

We ended up doing a combination of 1 and 2. We built features (product recommendation, sentiment analysis, etc.) to help people figure out what to buy then expanded our coverage into appliances, home and garden, babies and kids. We looked for growth by expanding how often we would be relevant (more categories, cover more of the buying funnel).

Most of our traffic stayed in electronics and our experience became thin. We were in a lot of categories trying to do a lot of things. I would've liked to see us do option 3, double-down on when to buy. That was the part of the market nobody else was playing in. Get that experience to be amazing then roll it out into other categories.

Lesson Learned:

Expanding your target market, like we did by supporting more categories and building when to buy features, doesn't help you get to market-fit. In fact, I'd argue it makes it harder because there's more needs to satisfy.

Said another way, is it better to start with a small group of people who love what you're building or a large group of people who are indifferent? I think you'd rather have the small group that love you. It's easier to turn a small group of rabid users into a large group than a large group of indifferent users into a large group of rabid users.

Keep your product narrow and focused on the small group of people most likely to love what you're building.

Something positive

I don't want to end this negatively because there's a lot we got right. I just tend to learn more from failures than successes. Maybe I should write another post about worked for us. For now, let me tell you about my proudest moment at Decide, when we lost almost all our revenue.

When we created the first version of our mobile app, we put Amazon's prices next to everyone else's just like we did on the web. Apparently Amazon doesn't allow you to do that on a mobile device but it's ok on the web. They want you to use their app to check their prices.

We received a notice from them informing us we weren't compliant and unless we removed it they'd suspend our affiliate account. We weren't making a lot of money but that account probably represented more than 80% our revenue.

We had 2 choices: (1) Comply and remove Amazon prices from our mobile app (2) Not comply, keep the prices and lose our biggest revenue stream. Proudly and with very little discussion, we decided not to comply. Most people want to buy from Amazon. Not being able to show those prices to our users wasn't acceptable. In the 2 years since we got that notice we never complied. When faced with a decision where we had to choose between revenue or our users, we chose our

users. I couldn't have been more proud.

If you want to see what I'm working on now checkout CoffeeMe, or as we like to call it Tinder for professional networking.

Original article here: https://medium.com/@hsukenooi/postmortem-of-a-venture-backed-acquired-startup-d74efec8b1a8

Startup Failure: Shnergle

Shnergle Post Mortem

I led Shnergle. It failed. The buck stops here.

I had bucket loads of life experience, but I was naïve at starting businesses; I was well outside my comfort zone (though would freely admit to it). Through the help of a lot of advice and online blogs (especially yours Mark Suster) we got a lot right, but we still made a bucket load of fundamental errors that I would tackle differently next time. A number of people put their faith in me (and their cash) and I feel I have let them down; the only saving grace is that the cash amounts were relatively small (courtesy of the government SEIS scheme) and we have all learned a bucket load in the process.

In consultation with the team I have pulled together a comprehensive 'warts and all' post mortem of what we got right and what we got wrong. It is not short (16 min read) as there is a lot to cover and I don't want to do it in too cold and dispassionate a manner.

Team

The importance of a Co-founder. You need to be a stronger person than I to start a business on your own. Having someone who is in the same boat as you to bounce things off and share the journey with is critical. Plenty of blog posts have been expended on the importance of having complementing co-founders; whilst this is critical, that complementing can happen in a number of ways rather than just the 'standard' one coder plus one business type; real life isn't that neat. In the spirit of 80% now is better than 100% late, work with people whose character compliments yours and with whom you can endure stress. Jonny and I have a lot of shared history from the Army and inherently 'get' each other. We know we always have each other's back but we aren't shy about robustly challenging each other. I am the accelerator and he is the brake. We both know how to lead and how to follow; if you can't follow you can't lead. Success!

Hire for character and attitude, flexibly scale up cost (salary) with competence. I still don't know how we did it when we hired Adam, Stian and Harshita, but we hit the jackpot. They learned fast, worked crazy hard, and asked for very little in return. When we started they said 'we've never made an iPhone app before, we don't know if we can do it'; I replied with 'don't worry, I've never built a company before so we're all learning here!' Considering it is the first mobile app any of the team have built I think it is a pretty awesome first effort, and was produced and iterated very fast. Success!

One of the Founders must 'get' the technology. I built my first PC when I was at school. At Uni I studied Engineering before I dropped out to join the Army; as part of that I learned basic computer programming (in C). I used to be heavily into computer games as a kid and would script some of my own levels. Although there has been massive skill fade through my time in the Army and banking and my attempts to get back up to speed on Codecademy have had limited success, I fundamentally understand the basics behind how computer software is built. I am never going to be able to build it as quickly and as well as a hard-core coder like Adam, but I can at least have a reasonable conversation with him about what we are doing and not get completely lost. Success!

Concept

Don't do unqualified product picking – I am an entrepreneur. I have led soldiers in 2 wars. I have an MBA. I can (relatively comfortably) manage massive personal risk. I am NOT your average consumer; I think very differently. I should NOT be a (consumer) product picker but should conduct more ruthless experiments to test concepts. Likewise we thought we had validation of the student market from our developers; we didn't. They worked crazy hours for very little return, based purely on pursuit of a vision; that is NOT standard student behaviour. Fail!

Importance of a clean message. Distil your idea down to a simple message that a 5 year old would get in under 3 seconds – that's about how much attention people are going to give you. This should be done before anything else. We only got to that stage just before we launched, and even then it could have been a lot better. Considering I have been taught and used Pyramid Story Telling in the past in consulting roles, I have no real excuse here. Fail!

Don't be too quiet with idea – No-one can match the passion, determination or depth of thought of a founder so don't be shy about being public with it (bin all that NDA tosh). If it's a good concept it will amass following; if it doesn't, why not?

Build a community before you even start the company. We were far too secretive. Fail!

Start with the market.

Speak to potential customers and people with key industry insights face to face to shape your initial concept, but then use social media to ruthlessly test your assumptions in the market before you build or finance anything.

- Facebook provides some incredible (if somewhat hidden) market research tools. If you want to find out about market demographics and sizes (in terms of people) start setting up an advert and playing with the targeting options – it shows you how many people in a given segment; iOS or Android first for example. It's not perfect but it's better than anything else out there you can find for free, and it's very quick to do.
- With the exception of a minuscule number of products that 'go viral' with instant product/market fit, almost all have to pay to 'acquire' users. Considering most people don't like conflict and won't give you honest (if negative) feedback face to face, you can kill two birds with one stone. Setup some Facebook adverts for your Facebook page and get an idea of Customer Acquisition Cost up front.
- Most people get notifications from Twitter when they get a new follower. Sure some will have turned these off or may ignore them, but take out a very cheap (£7pm) ManageFlitter subscription and use your 'product' Twitter account to follow the followers of potential competitors – you can analyse how many follow you back.

We spoke to a lot of bars and merchants, but we didn't try any Facebook ads or Twitter stuff until pretty much launch. It was ultimately an excessively high User Acquisition Cost which meant the company had to shut down. Partial Success.

Inaccurate Market positioning. We thought our app would be most popular with students. It was actually most popular with professionals aged 25-40 (User Acquisition Cost was 85% lower!) but they are fixated with social concerns over and above saving time. Whilst professionals want to save time, they are not nearly as engaged with mobile apps, creating a critical demand problem (we had 28% of users return after their first day, and 8% returning 4 times or more). We could have learned this much earlier with Facebook Ads. Fail!

Are you well positioned to build a community quickly and cheaply? If your idea relies on virality or massive uptake and you/your primary social circles aren't massively active on social media, you will have a hard time building a community and are hugely unlikely to get any kind of virality. A small core of our friends were huge advocates, but the majority completely ignored Shnergle. It's not personal; they just weren't buying what we were selling….even for free! Fail!

Does your idea only monetise at scale? If your idea can only be monetised at scale, head to San Francisco / Silicon Valley. There isn't enough risk capital, or enough risk appetite, in the UK/EU venture market to pour capital into unproven R&D concepts. If you want to build in the UK, find some way of charging money from day one. You can still use a freemium structure to up-sell later. Shnergle was never going to monetise before it had scaled fairly significantly. Fail!

Financing

Fundraising is a distraction – make it quick. We were not from Tech Startup backgrounds so didn't feel conventional angels were a viable option. We raised our initial equity capital (£75k) in three weeks by breaking it down into bite size chunks (standard buy-in was £5k) and going to people who knew us the best and who knew that when we say we are going to do something, we do it. Courtesy of the government SEIS scheme the standard amount at risk was £2.5k, pretty minor considering the opportunity we thought we had, and the cost of your average well paid middle class holiday. In some cases I had to caution investors from piling in too much cash so as to properly manage their risk. Success!

Avoid legal costs from financing rounds. Don't bother with Option schemes or shareholder agreements until raising institutional capital as they are a waste of time and legal costs to setup; they will also need to be changed if you get institutional investors on board. Letters of subscription are the way forward and cost next to nothing. Everything has to be done on trust and kept simple so you can focus on all the other clearly more important parts of the business. Success!

Don't try and use a 'valuation' when pulling together your first cash. Any valuation model that is not based on a firmly quantifiable growth trajectory, or better still robust cash flows, is highly questionable and subject to massive margins of error. VCs are familiar with the dark arts of trying to pin the startup tail on the right part of the valuation donkey, but you are not. In due course you can start to get a feel off other entrepreneurs and investors as to what the market is likely to value you at, but to begin with that is all rubbish. Work out how much cash you think you need, and how much equity you

are willing to give away; make sure there is plenty of room for outsize rewards for those taking the early risk, but that you don't dilute yourself too much before you tackle your proper institutional investment rounds. We worked on 10% in exchange for £75k (thus valuation at £750k – pretty much an upper limit). In the end we sold 7.66% because I invested a load myself as part of our £75k. Keeping the founders at 92.33% would have given us plenty of flexibility if we were actually able to raise a Seed round. Success!

Accelerators – Yes or no? We looked at a number of accelerators – WAYRA, Seedcamp etc. We decided to avoid them based on a few factors:

- We got the feeling there was a lot of pitching and a lot of chatting with 'mentors', but we couldn't really see how those mentors would be much use dropping into our business if they didn't really get a good understanding of it. This seemed like a lot of friction and distraction to us when frankly we could read up on it online. Think full time MBA vs. MOOC as an analogy.
- From our research the investment terms seemed pretty standardised for each company, and we could get better ones with our actual investors.
- The accelerators appeared to come with a lot of hype and publicity that we didn't want to be exposed to too early; we wanted to stay under the radar for a bit.

In hindsight I don't think it's a clear cut case whether an accelerator would have been a good idea. I think we may have learned more by making mistakes in isolation than if we had just 'done what we were told' in an accelerator. Maybe the hype and publicity would have helped our community problem? Perhaps we would have made mistakes faster in an accelerator and been able to fix them? Undoubtedly our networks would be stronger if we had done an accelerator. Neutral.

Don't waste people's time. We have been lucky enough to be introduced by email to a number of prominent VCs in London, one of which was even good enough to take a non-investment meeting just to chat through advice. There were very few bites when we first reached out (see 'Importance of clean message') so we waited to try and build a compelling case before reaching out to start our Seed Round. In the end we didn't think there was a compelling case so didn't waste the VC's time trying to pitch them, we shut down instead. Hopefully they will respect us more for this and we can build a relationship ready for next time when we do this better. Only time will tell. Success! (Of sorts)

Don't underestimate the costs of business. The barriers to entry to internet/mobile businesses are very low, resulting on huge barriers to success due to crowded markets. Don't under estimate the costs of marketing your product, both in promotional spending and marketing staff, growth hacking etc. This all needs to be paid for either with revenues, investment or both. If you think you are going to cover it with Investment (such as VC), there had better be a compelling case as to why it is basically printing money for them. Fail!

Keep it Lean. Any money spent on testing a concept is at risk of being money wasted if that concept doesn't work out, so spend as little as possible. Professional Coders and designers might be able to get stuff going in their spare time for free, but if you're not one of these, and you don't have a network of them, you are going to have to put some time and cash into getting things going. We got a company from slides to market, with 5 people working near on full time for 6 months, with no prior tech company experience and zero tech networks, all on £75k; less if you factor in that there is still a bit left. I think that is pretty damn lean considering we were all having to live off our Shnergle Salaries! Success!

Marketing

Market to people when they are receptive to it. We tried loads of different ways to market the app, but fundamentally the most cost effective ones were when people are receptive to the marketing:

- Inbound 'content' marketing. Videos are most effective inbound marketing, followed by imagery, then blogs. Good blogs can take ages if you are bad at them (me), as do videos. Focus on photos/imagery initially. We found Facebook to be massively more effective than Twitter for content distribution. Inbound marketing worked well for growing our Facebook following, and our Facebook following was best for getting downloads.
- Promoting to people on nights out in bars. Attractive promo people cost money and although you get good qualitative feedback, you don't get many users. Even without the promo people it still takes time. People are just not that receptive when on nights out.
- Materials in bars (flyers, posters etc.). We know we got a fair bit of referral traffic from these from measuring QR

codes, but not enough to justify the cost and fundamentally they didn't materially increase our user base.

- Pitching merchants – individually and en masse. We had loads of positive feedback, but without the users they really aren't interested. The 2 bars that did become official venues only became venues because we were good customers and they wanted to keep us happy/help us out.
- Flyering at freshers fairs. We were able to build significant awareness this way, but it was time consuming and resulted in precious few user installs; we were pretty glad we didn't pay the rip-off fees to host a stand but instead either snuck in gorilla style or flyer'd outside.
- Wristbands. These worked well for a short period of time – lots of people wanted them, but then they got home, took them off and their value became limited.
- Media. We got some great press coverage – a half page feature in the Evening Standard, top apps of the week in Gizmodo, and a load of coverage from getting the guys at the front of the iPhone 5S launch queue to wear our t-shirts. The evening standard article got us a 21% jump in users over night, but considering the distribution and number of impressions of that story it was very small in absolute numbers. It was also not easily repeatable.
- Facebook ads. Mobile App install Ads in user's timelines work more effectively than anything else we have tried. As they are completely measurable they can be optimised and tweaked to drop User Acquisition Cost over time. If I was ever to market a consumer product again I would spend most of my outbound marketing budget on Facebook Newsfeed Ads.

Success! Eventually…. after many wrong turns….

Facebook is a double edged sword. You need to create a profile on App Centre in Facebook if you want to run Mobile App Install Ads in Facebook. This is great for discovery, but has one real pain in the ass feature – public Monthly Active User Count. This is great once you are up to scale as it aids transparency and attracts users based on the herd mentality, but it really doesn't help when you are starting out : 'Hmm so this crowd sourced photo app that I see in my news feed has only got 300+ Monthly Active Users! I doubt that's enough to make it work, I won't bother'. Cheers Facebook! See 'Don't be too quiet with your idea'. Partial Success.

Understand the value proposition of your website. If your business is through a mobile app and not a website, don't waste valuable engineering talent on your website – it is just a marketing channel. We built ours in Adobe Muse for almost zero cost with no real technical knowledge required. We could always have built a better one with Twitter Bootstrap later if we needed to. Success!

Don't get carried away with Product Videos. Don't make a product video until you have a product and then keep it simple as your product will change. If you focus on the vision rather than the detail then you can make it in tandem to your product. Our videos were very helpful at explaining our offering, but we probably should have been more ruthless and only had one video – see 'Importance of a clean message'. Partial Success.

Get your head round the concept of Growth Hacking and data analysis. Once launched we were ruthless about analysing Shnergle's limited data to try and understand what was happening. We tried lots of experiments to improve results and measure what worked best. Unfortunately the concept was sufficiently flawed that we couldn't make a great deal of headway in this area. Partial Success.

Product

Build a seeding product not a final vision. This is the single biggest product execution mistake we made, but we thought we had seen it and avoided it. How did Hailo have so many black cabs available when it launched its app?

- It first released a driver app that provided a social network for drivers, helping them analyse their activity
- Three of the Founders are former cabbies so they were able to sell it to the community as a fight back against Addison Lee

We thought we had mimicked this by allowing merchants to setup accounts from the word go and getting the email addresses of almost every bar in central London prior to launch. They were going to setup immediately and start creating content for the inevitable onslaught of users. [Tumbleweed passes]. From over 3,000 email addresses we got precisely zero response in the form of advance signups, or downloads on launch. We got a bit of web traffic, but it was clear they wanted the users first (if they were even interested in the concept).

We tried to fake it by racing around London ourselves adding content and getting our good friends to do the same, but it wasn't enough. We compounded our own problems by putting an 8 day limit on how long content would show in the app on the basis people wouldn't want to see older content than that. To be fair, no-one did want to see content older than 8 days….but they wanted to see something, and the 8 day limit meant the app often showed lots of 'No activity in 8 days' icons. This was a colossal error. We should have built some kind of service that got all the Bars using us and uploading photos long before we build the Shnergle app for consumers. EPIC Fail!

Native, Hybrid, WebApp, Responsive Web etc. Make your Native/hybrid web decision based on when/where you expect your customers to be using your product, and to balance risk/time/cost. We assessed that customers would be using the app in a hurry whilst on the move, so a slick, highly responsive app utilising a number of native APIs was required. Considering the uncertainty around cross platform development, it therefore became a no brainer to go fully native. Success!

UX is critical – focus on one key functionality. The focus of our app was real time location based photo sharing and viewing. Although we worked with an excellent UX designer (Steve Blyth) we tried (I think against Steve's better judgement) to add too much into the app. We had RSVP'ing, Comments and a whole load of other unnecessary elements that we wasted time designing and building, slowing product release. Although we did well with an intuitive design, we should have kept it a lot simpler. If we hadn't wasted as much time on unnecessary features we would have been able to polish the app much more too. Neutral

Instrument Everything. As our eventual Business Model was going to be based on Data Analytics and we were using Facebook login exclusively (which gives you loads of demographic and use data) we thought we had already instrumented everything to the degree needed in our own systems. We had about 90% of what we needed but it wasn't well tied together or easy to dive into. We should have just used Flurry from the word go and tied events to absolutely everything. Treat the data you need, and the data you are going to provide customers as different beasts, at least initially. Partial Success.

Keep login simple but provide choice. Even Facebook said at the FB Start event that you shouldn't just use a Facebook login, but should offer users the option of an email login alternative. In this scenario Facebook claim conversion rates from download to user can be as high as 70-80%. Although we had a slick login flow from only using Facebook, our conversion rate was 59% so this could have had a material impact if we had included email too. Twitter login is a complete waste of effort though – we looked at it and decided against. Considering the number of our users who actually added twitter accounts (2.3%) this was a good call. Partial Success.

Take as little Facebook data as possible. Only take the Facebook data you actually need and only take it when you need it, then people are more likely to agree to it. That is exactly what we did, hence having quite a good conversion rate for a Facebook login only app. Success!

Understand the friction of the Apple app review process. Apple make great experiences, but they are a pain in the ass to deal with. I agree they should review apps to conduct quality control, but they seriously need to get a grip when reviewing app updates. It's not exactly clear what they test, but they will reject it if you have the audacity to mention (for example) 'Google', even if the context is transferring from Google Maps to Apple maps. That is pathetic. When initially submitting the 'In review' took about 2-3 days for us; when reviewing updates this dropped to 5-12 hours. It really wasn't clear what they were doing for all this time as we didn't get any activity on our servers during update reviews, only on the initial review. For both circumstances the 'waiting for review' period was about 6 days. Unsurprisingly this jumped to over 2 weeks over the iOS 7 launch. Neutral

Conclusions

Congratulations – you made it to the end!

I can honestly say 2013 has been on a par with my first Iraq tour (it was a particularly rubbish 6-7 months when I was 22-23) for the shaping effect I suspect it will have on my life. Everyone in the team has learned an incredible amount in a fairly compressed time period, and given our new networks and experiences we should be able to do the next startup more efficiently, more effectively and ultimately more successfully. I am sure most of what we have learned will seem obvious to those already in the tech sector, but hopefully we have just provided some tips for anyone considering leaping into Tech startups in the near future. If nothing else we have provided an effective way of recording our lessons learned

both for our own team, and for our Investors.

Original article here: http://www.shnergle.com/blog/2013/10/23/shnergle-post-mortem/

Startup Failure: Tigerbow

Tis the Season for a Tigerbow Post Mortem

Every holiday season, folks reach out to talk to me about the gifting market because of my first startup. It was called Tigerbow, and it let people send real gifts to email addresses (and Facebook profiles, Twitter accounts, mobile phones, etc.). I poured my heart and soul into it for 2-3 years, and it failed. Facebook's (re)launch of Gifts and the holiday season got me thinking that I ought to finally do a post mortem. It's been a few years and another startup, so forgive my memory, but here goes…

Let me first dispense with some of the more obvious insights:

Founders need to be full-time. If you're not full-time, you're not a founder. Until you are. (Duh.)

You need full-time tech talent. You can outsource some stuff, but you're going to rebuild it. Learn to build a product, code a little or find a tech co-founder. (Duh again.)

Market timing is critical. We tried raising money for our company during the financial crisis of 2007-08. Needless to say, we were running up against stiff headwinds. (Duh?)

Execution is way more critical. I cannot attribute our failure to market conditions. We were unable to execute on a strategy to drive early traction. (Duhhhh.)

Don't raise money from people who don't invest in startups. We raised a (comparatively) small amount of money from friends and family. For the most part they were very supportive, but there were exceptions.

Aside from the fact that we got little (non-monetary) value added from these investors, people who are unfamiliar with investing in startups and the risks and challenges of building a company will drive you bananas. (Tempting, but don't / duh.)

Too early is wrong. After Facebook's acquisition of Karma and subsequent relaunch of its new Gifts product, a handful of people told me how "smart" and "early" we were with Tigerbow. But too early is wrong. Early / good timing plus building great products and solid execution are right. (Some people will think you're smart, but who cares. Duh.)

And now some more things we learned that were a bit more specific to Tigerbow, ecommerce, gifting, etc.…

My Awesome Math. (It's awesome because I made it up.)

The gifting market is indeed large. We estimated that more than 20 percent of merchandise purchased was with the intent to give that merchandise to someone else (i.e., a gift). That estimate was predicated on the idea that online retailers see 40 percent of sales in the fourth quarter (so about a 20 percent bump compared to other quarters). Assuming that entire bump is holiday gifting and that there is no additional gifting during the year (wanna know what the biggest gifting event each year is? hint: it's birthdays) gives you 20 percent.

Yeah, it's kind of a BS analysis, but it sounds good and, luckily, compares well to other statistics, so I'm sticking with it.

The problem we were trying to solve, however, was when someone wants to buy a gift but doesn't because they don't know how to get it to the recipient (i.e., there is shopping cart abandonment). Stats will show you that shoppers abandon carts something like 60 percent of the time, and one of the top reasons, though not the top reason, is because checkout is too complex. Figuring we could increase the bottom of the retail funnel by streamlining checkout, we would bring retailers more money.

But how does a user know that this feature exists until they reach checkout? And was it a big enough issue? And was it a big enough issue for retailers to focus on it at the expense of focusing on other issues? In retrospect, the answer was no (at the time), but to figure that out we tried a bunch of stuff (that I'll describe in reverse chronological order because I'm feeling acrobatic…)

Partnering with Retailers.

After one of the first times I presented Tigerbow at an event, someone walked up to me and suggested I reach out to Amazon for an integration opportunity. It was at this point that I became retroactively mortified for every piece of "advice" I ever delivered to former banking clients and investments. Really? I should think about partnering with Amazon? Well, gee, why didn't I think about making that call? Grr.

Of course we thought the best course of action would be to integrate our technology (a double blind set of databases matching a virtual address with a physical address using a key that only we knew) into merchant checkout. First thing to learn: merchants do NOT screw with checkout. This is how they make their money, people. No matter how good your clever plan sounds to help them increase the bottom of the funnel and generate more revenue, they are extremely reluctant to reconfigure any onscreen real estate, let alone implement a new technology or order flow. This one was pretty obvious in retrospect, but you'd be surprised how many companies believe they've got a solution for solving a checkout problem that retailers are dying to implement. There's a reason you only see credit card, Paypal and maybe a few other checkout options from extremely powerful companies like Google and Amazon. And even they have had significant trouble penetrating here.

Second point on this (and a corollary): retailers are way more focused on the top of the funnel (getting traffic to their site) than the bottom. They know their conversion metrics, and there are simple, well-known and inexpensive ways to increase the top of the funnel (e.g., affiliate deals). So we knew we would have to prove our value-add elsewhere before approaching big retailers.

Partnering with Dating Sites (and the unfortunate discovery of the "freenis.")

We had already built a rudimentary (Atari 2600-looking, and not in a good way) online store that I'll talk more about in a bit. We hustled and got some decent press (including my favorite, a headline from Mashable: "Big Idea or Just Creepy?") so we had some people visiting the store and buying books and flowers and such. I'll spare the details of how we figured this out, but we noticed that we were seeing some Tigerbow gifting via Second Life. (Remember them? They still exist – swear.)

Aha – seemed like a natural fit to integrate our platform into dating sites so that guys could entice dates by sending real flowers rather than some pixels. (Not something I would do, but I also wouldn't wear a freenis – I promise I'll explain the freenis soon.) We had some (very) limited success integrating our platform into some dating sites such as Passions Network, a collection of 100+ niche dating sites, including what I believe was the largest Native American dating site as well as a niche community for (and I swear I'm not making this up) lesbian amputees, but the harsh reality was that gaining incremental revenue from or enhancing the user experience with real gifts wasn't compelling enough for most online dating sites to make any meaningful changes to their properties.

They could more easily use that real estate for additional advertising or other engagement opportunities.

So we determined the only way we could truly present our unique value proposition was to build out our own store, which we did… before we learned the definition of a freenis. Without further ado, here is that story:

On April 18, 2009, I received an email from my co-founder titled: "dude [Second Life] is weird." We both knew Second Life was weird, so for him to feel compelled to elaborate on this was alarming. His email directed me to this guide for getting started on Second Life. The guide includes 30+ things every newbie should know, including this gem:

"One for the guys: don't bug every female you see for sex. Especially don't attach your 'freenis' (freebie penis) and run around everywhere. You WILL get laughed at."

Freenis. Now you know.

Building an Online Store.

My co-founder started the company by building a very basic online store that allowed people to send books as gifts with only an email address. You picked a book, entered an email address and a message, and your recipient would receive a message asking them if they wanted to receive or reject the gift. If it was rejected, the transaction unwound (no harm no foul), and if the gift were accepted the recipient would enter shipping details (which were kept private from the sender), and the gift would arrive.

We started with books because they make decent/easy gifts for any occasion, they're relatively inexpensive, there are drop shippers like Ingram that will allow you to integrate and fulfill orders easily and, well, it worked out pretty well for Amazon as a starting point.

Books worked out OK for us, too. In fact, the only hitch I recall was when I was testing sending a book to my then girlfriend (now wife – bless her heart). I selected a book that would make her laugh when it showed up (recipients didn't necessarily know what they were receiving). The book showed up, and the test was a success (seemingly).

It wasn't until we were at dinner with my co-founder and his then girlfriend (now wife) when my brave soulmate asked me what on Earth she would do with the children's bible I sent her. Evidently she didn't think that book was that funny. And neither did I because I sent her a book of lesbian erotica. When we realized that the gift note that accompanied her children's bible was from an aunt to a nephew we were very curious to see the reaction on the little boy's face when he got his collection of lesbian erotic stories. Can anyone say world's greatest aunt? (True story.)

We added flowers because they're one of the most obvious gift categories (and matched nicely with our dating partnership strategy) and because we found a drop shipper called Grower Flowers that was willing and eager to have us integrate our platform. Evidently they had tried something similar with Plenty of Fish a while back. And evidently we and they weren't the only ones to try this real gifting thing. Jared Polis, who founded Bluemountain.com and ProFlowers – both of which sold for a bajillion dollars – before he became a congressman, started FrogMagic, which did essentially the same thing as Tigerbow. Who knew.

It became clear pretty quickly, though, that we would need to build a full-fledged online gift store if we wanted to prove our value proposition. That would require a full catalog (which we tried to sneak by via adding things we could do ourselves like greeting cards and affiliate deals with select retailers that we could fulfill manually), a spectacular user experience and prolific marketing (all of which required lots of money, expertise and resources that we didn't have). And still, it would be extremely difficult to convey to people that they could utilize our key feature (yes, feature) – the ability to send a gift to someone's email address.

In Conclusion…

Start a company with a simple idea. But also remember that you absolutely must have a simple, clear and effective way to build early traction and demonstrate your value proposition to derisk your project for potential investors. Otherwise, it's like running around everywhere with your freenis attached – you might not get laughed at, but you will fail.

Original article here: http://www.alleywatch.com/2013/12/tis-the-season-for-a-tigerbow-post-mortem/

Startup Failure: Flud

Why Startups Fail: A Postmortem For Flud, The Social Newsreader

In late July, news broke that Flud, the social news reader for iOS, Android and Windows Phone, was headed to the deadpool. Startup failure is an all-too familiar, even cliche, story in Silicon Valley. But when San Diegoan co-founders Bobby Ghoshal and Matthew Ausonio officially shuttered the Flud app and website in August, after three years of development, it was a disappointing final chapter for a product that had attracted more buzz in a year than the average startup sees in its entire lifecycle.

It was also a surprising conclusion, considering that Flud had managed to raise $2.1 million in seed financing from investors Detroit Venture Partners, Ludlow Ventures, Behance co-founder and CEO Scott Belsky and JFJF Ventures, among others. Furthermore, reports circulated last year that FLUD was continuing to see interest from investors and was in the process of raising up to $8 million in new funding.

But Flud never closed on the round and today, the startup is no more. In fact what eventually killed Flud, was that the company wasn't able to raise this additional funding. Despite multiple approaches and incarnations — including a pivot toward enterprise — in pursuit of the ever elusive product-market fit (and monetization), Flud eventually ran out of money — and a runway.

To date, the founders have yet to speak publicly in any detail about Flud's closure and why the company wasn't able to turn a promising start into fame and fortune. Last week we caught up with Ghoshal, who graciously agreed to peel back the curtain and offer a bit of a postmortem on Flud– for sake of clarification and, of course, edification.

Startups are hard — don't let anyone tell you differently — and everyone fails. Some just learn to fail better than others, and that's what makes the difference.

Flud's story has much in common with another recent casualty in the mobile space: Sonar. In a recent post on Medium (which is a must-read for all entrepreneurs, investors and startup peeps), former Sonar co-founder and CEO, Brett Martin, offered insight into why his own mobile, location-based startup went bye-bye.

Sonar and Flud were both downloaded by millions of people, were promoted by Apple and Google, raised multiple millions and received high praise from the media — and yet you know how both stories ended.

Demand

As Paul Graham has famously said, one of the keys to building a successful startup isn't necessarily having a brilliant idea, but to "make something that people actually want." When Ghoshal and Ausonio began developing the idea for Flud in the summer of 2010, their big vision was to build a product that wouldn't just be another RSS-based news reader or social magazine, but an actual social news ecosystem (and a design-centric, i.e. visual, one at that) — an "Instagram for news," if you will.

While it's easy to say, as an armchair observer, that part of the reason Flud failed was because the market was already saturated by similar products and the competition was too strong. After all, when Flud launched its first app for the iPad in August 2010, Flipboard and Pulse were already well-established, had millions of users and mostly dominated mindshare in the newsreader market.

Plus, Flud had to work against the fickle, uncertain and ever-evolving relationship people have with news and news consumption tools. They always say they want a better newsreader, are willing to try something new, but then they cling like pocket lint to what's familiar. Look at what happened with Google Reader: Most digitally-savvy people seemed to agree that, while Google and RSS were fantasti-awesome for many years, eventually smarter, real-time tools offered greener pastures. Seemingly in agreement, Google started dismantling the service in 2011, finally turning it off this summer, and, of course, the Web went bananas. It was as if Google had just turned off their electricity and punched their mom in the face.

But, really, when Flud launched its iPhone app in December 2010, there was plenty of demand for a "better" way to consume the news — in truth, there still is. As an illustration of this, when we talked to Ghoshal at the end of December,

the app was "adding a new user every four seconds."

It's also why (at least in theory) Flud struck a chord with the media: Fast Company put its iPhone app among the "Best UI Designs" of 2010 and NBC dubbed it "the future of news." Hell, even some guy named Rip Empson (I know, right?) said Flud could establish itself as a "force to be reckoned with" in the news arena.

In retrospect, it's pretty apparent that we were, in a word, "not exactly correct." (Hey, even the best get it wrong sometimes, right guys? Right? Hello?) Nonetheless, and of course with a little of my own, requisite defensiveness, the demand for a better newsreader was there. And, going one step further, I'd argue that the initial vision behind Flud — the vision of building a full-service, design-friendly social news ecosystem — had appeal. And still does.

Just ask the big social networks. Today, Facebook is in the process of developing its own social newsreader as it inches into territory where Flipboard, Pulse and Feedly are arguably still the best options, if not the best-known names. In regard to Pulse, LinkedIn bought the popular newsreader for $90 million earlier this year to help it develop its own content hub and news reading strategy. Not only did LinkedIn think it was worth $90 million, but has kept the apps alive in spite of the acquisition (and overlap) and continues to update them.

Competition

But who cares about them? (Yes, that's half a rhetorical question.) In his post on Medium, Brett Martin details the results of Sonar worrying too much about competitors, like Highlight, Banjo and a host of others. Like Sonar, Ghoshal says that there was always noise about how well Flipboard and Pulse were doing, but it was always more psychological. In the end, you can't let what other companies are doing drive the direction of your business — the more you let them into your head, the worse it gets.

It's impossible NOT to pay attention, naturally, but your response should be "Hrmm, that's interesting — not 'Oh, F#$%^&*'" … Know a good idea when you see it (a "good" idea being a better way to solve the problems your users are having NOW, and if you want to push it, the problems they'll be having in a few weeks, but NOT the problems they'll be having in a year, that's hubris. You're not Steve Jobs), but don't let it lead you away from your overall vision.

Instead, if a startup is lucky enough to hit on an idea or a product that users actually want to use, what's far more important than what the competition is doing is understanding what customers want and narrowing the gap between what customers want, what parts of the product they're actually using and what your product's functionality/feature set as it exists at the time.

Press

Don't waste too much time worrying about press and coverage. If you build a great product, press will come — take it from me, I'm a doctor. Then again, Ghoshal tells us, believe in your product and vision enough so that you're not surprised by press if it does come, and so that you'll be ready.

Ghoshal says that, leading up to Flud's launch of the app three years ago, he and his co-founder were working on it part time. Hindsight is 20/20, as they say, but in part because they didn't expect to get any press and hadn't done a press-fueled launch before, little problems at the beginning snowballed quickly.

Flud was lucky enough to get a ton of press around its early product launches. However, from then on, when the team would launch a new product, they would reflexively go after press, Ghoshal says. They would get press, users would overload the app, the system crashed, and when they finally got it back up, there were still too many bugs left, and, in a world of eleventy billion apps, people tend to be pretty unforgiving of glitchy user experiences. (As they should, by the way.)

The Flud CEO then concluded, chuckling: "One of the most important things I learned," he says, "is that when you launch a new product — especially if it has some real 'novelty' to it — the best thing to do is shut up about it."

Test, Test And Test Again

"We really didn't test the initial product enough," Ghoshal says. The team pulled the trigger on its initial launches without

a significant beta period and without spending a lot of time running QA, scenario testing, task-based testing and the like. When v1.0 launched, glitches and bugs quickly began rearing their head (as they always do), making for delays and laggy user experiences aplenty — something we even mentioned in our early coverage.

Not giving enough time to stress and load testing or leaving it until the last minute is something startups are known for — especially true of small teams — but it means things tend to get pretty tricky at scale, particularly if you start adding a user every four seconds.

Ghoshal also says that what testing they had done was too narrow in scope: "We also hadn't tested our assumptions on what people would care most about in a newsreader — we knew that they'd want to share with friends after reading something, but we didn't think they cared about things like influence. They did, and by the time we added that, some people had already moved on."

This relates to the advice above: Know what your customers want and how they're using the app, and as Brett Martin says, optimize for actual user behavior and, as much as possible, focus on removing friction from their existing behaviors. Not focusing on more abstract things, like tech or an amazing set of APIs.

It's Hard To Recover Once Things Start To Spiral

Flud was too concerned with press and not concerned enough with testing new features beyond a few trusted people, he says. And, after that, "things spiraled pretty quickly, because our new users heard about the bad experience from early users, reviews didn't help, so we lost a lot of mindshare."

Even after Flud added much-needed features like email notifications to its early product, it was too late, he said. "It's really hard to recover from that early spiral." In response to these early problems, the team went heads-down to focus on rebuilding the tech side of Flud, while its competitors focused on user acquisition, the CEO says. But this was a reflexive decision, and it led to Flud going too far in that direction.

As Martin says of Sonar's experience, for startups, there's an ongoing, internal battle over whether to focus on growth or engagement, and how to balance the two. However, if a startup is building a social product, he says, growth is what's key — engagement doesn't matter. Ghoshal's account echoes that of Sonar — perhaps not surprising considering Flud was essentially trying to reimagine what a social network should be in the context of news consumption.

In the long run, the Flud CEO believes that the team did eventually succeed in building a much better product on the tech side, but when it came time to raise capital, "investors don't want to invest in a third-tier newsreader." To Martin's point — all they cared about was growth. And because of it's early struggles, Flud had a steeper hill to climb thanks to plateau-ing user adoption.

Figuring Out When, And How To Pivot (Listen To Feedback)

Once Flud saw that investors were balking at its consumer product, it had to take stock in the business. The team quickly realized that money was getting tight and it had to move quickly to find a new market where customers would pay — in other words, where it could build a real business.

Early on, Ghoshal said, a number of enterprise players had reached out to the company, saying things like, "even though Flud is a consumer product, our executives just bought iPads, and they want to use the app, but they need to use it behind a firewall." Essentially, these companies wanted Flud to build a white label product, keeping the user experience the same, but optimizing it for "big company" use cases.

At the time, Ghoshal says, he didn't see the opportunity, so Flud turned down those opportunities. But when it came time to pivot, they went back to those initial leads and found the demand was unchanged. Flud moved quickly to launch a new enterprise version, which launched in December of last year with 1,500 companies on its wait list.

There's Always Enterprise, Right?

Impressively, in the four-plus months that its enterprise product was live, the four-person team was able to bring in about $500K in bookings. However, the team hadn't given itself enough time — it ran out time and money before it could really

get the ball rolling in any significant way.

In spite of Flud's failure, Ghoshal believes that there's still a huge opportunity for a "consumerized," social, news-sharing network for medium-sized and businesses. "The enterprise is hungry for a product like this," the CEO says, "and it's something not many are talking about. Big companies have a content sharing problem that Yammer isn't adequately solving."

These companies want to offer a product through which employees can read content about their customers, along with internal news and content shared by executives and managers. They want to be able to track what people are reading and offer better, more personalized content.

"There's a huge opportunity for a product that offers businesses, particularly enterprise businesses, a better way to read, share and syndicate content internally," the Flud co-founder tells us. While Flud began to do just that, it also ran into problems that any consumer-facing startup moving to target the enterprise has to deal with, and it didn't have the resources to see it through.

As many know, Ghoshal says, building a product for the enterprise is an expensive proposition and startups need to make business development hires, deal with the hand-holding and time required to negotiate the more complex integration processes in enterprise — not to mention the capital demands that come from dealing with longer sales cycles.

In the end, though it was seeing encouraging data from its early customers and had nibbles of interest from investors, "they all wanted to see three to six months worth of data, but we only had two months worth," Ghoshal explained. Investors wanted to see more validation, which Flud didn't have, and that, in the end, is what finally killed the company.

Moving On

The decision to shut down or sell a company is a nerve-racking, white-knuckled process down to the last minute. The process is all-consuming, and leaves many in bad sorts. While one would assume that it would be tough for employees and founders to move on to the next opportunity, once there's finality, it can be a lot easier to move on.

The whole Flud team has moved on to different projects, and Ghoshal says he's beginning to work on his next startup, though he's not ready to talk about what that is just yet. In the meantime, the founders are in the process of looking for a buyer for Flud's assets and are in conversations with several potential acquirers.

His advice to startups on what to look for in a potential acquirer? "I think for most, it's a matter of figuring out which company would benefit the most from your product," he says. "It's probably obvious, but the company that's going to give you the best value is the one that wants to sell the product right away … and isn't going to just give you cash and then shut it down."

Original article here: http://techcrunch.com/2013/09/22/why-startups-fail-a-postmortem-for-social-newsreader-flud-and-what-to-take-from-sonars-demise/

Startup Failure: Condom Key Chains

How My Start-Up Failed

There was no doubt about it: I had discovered The Next Big Thing. Like Edison and the lightbulb, like Gates and the pc operating system, I would launch a revolution that would transform society while bringing me wealth and fame. I was about to become the first person in America to sell condom key chains.

I first encountered the condom key chain while working in Bangkok. Faced with a warehouse full of soon-to-expire condoms, the ingenious leaders of a Thai community development organization took the aging prophylactics, sealed them in plastic and attached a key ring with a tongue-in-cheek logo: "In Case of Emergency, Break Glass." They couldn't sell them fast enough.

My belief that the condom key chain would quickly eclipse the legendary success of the Pet Rock was confirmed by a simple market survey. I showed one to my mother. "Robert," she said, "these are the funniest things I've ever seen! Get me 50. I'm going to give them to all my friends." Mom loved it. She thought all her friends would love it. America would love it. What more did I need to know?

Plenty, as it happened. Though I had a Stanford MBA and regularly consulted on multimillion-dollar projects, I didn't know the first thing about starting a business. When I asked a successful classmate how to invoice a customer, he suggested I go to one of the large business-supply warehouses. "They sell 'Business in a Box,' " he told me. "It's got everything you need." I didn't realize he was joking until I asked a clerk for one at Office Depot. While there, I bought a copy of How to Form Your Own California Corporation. I spent $30 on a special seal for all the important documents I would have to emboss. I registered a "doing business as" name. Finally, in anticipation of huge sales, I linked my marketing database with the word processor on my personal computer and invested $10,000 in my venture.

My order of 10,000 key chains was scheduled to arrive from Thailand just in time for the San Francisco International Gift Show. When the shipment came in, I raced to the airport.

"Where are your papers?" the clerk asked.

"What papers?" I asked.

"Customs clearance," he told me.

"Customs clearance?" I said. "I've got 10,000 condom key chains to get to the Gift Show by tomorrow."

"Well, Mr. Strauss, I guess this is where the rubber meets the road," he said, breaking himself up.

Hours later, with papers finally in hand, I backed up to the loading dock. My 10,000 key chains had been shipped in two cardboard boxes. Lifting them from the dock, I noticed that the bottoms of both boxes were discolored with large greasy stains, like the blotches beneath leftover pizza. With the Gift Show starting in less than 24 hours, there was no time to complain about mishandling. Once home, I ripped the boxes open like a kid at Christmas.

I had written a series of clever slogans to supplement the original "In Case of Emergency, Break Glass." There were 2,000 marked "Slippery When Wet" and 2,000 marked "Merging Traffic Ahead" adorned with little yellow yield signs. For Pac-10 schools, I had 2,000 "Beat the Trojans" key chains. How could this miss?

I reached into the boxes to fondle these jewels of schlock merchandising. My hands came away covered with light, clear oil. The key chains were all stuck together; my entire shipment was covered with goo. It didn't take long to realize what had happened.

The outdated, lubricated condoms had been sealed in plastic, and the change of air pressure during shipment had forced the lubricant out through the seams. With the Gift Show beginning the next day, my elbows deep in slimy key chains and my $5,000 payment deposited in Bangkok, I began to panic. I filled the bathtub with hot soapy water, dumped in my 10,000 key chains and began scrubbing.

My love affair with my product soon began to fade. The key chains would not come clean. No matter how much I scrubbed, they still felt as though a posse of banana slugs had just oozed over them. Two weeks later, a friend of mine flew to Thailand on vacation. Back with her went my key chains. A single American woman entering Thailand with 10,000 condoms. She said the customs officials were very accommodating.

The replacement shipment was slime-free. I soon began sales calls on buyers ranging from novelty shops and porno stores to gay rights groups and Planned Parenthood clinics—customers conspicuously absent from the cases I had studied at the Graduate School of Business. It became all too apparent that the condom key chain was no Pet Rock. My gross profit margin was tremendous, but overhead had driven me $13,000 in a hole that was getting deeper and darker. Retailers told me that what the customer really wanted was a key chain with a usable condom. Soon I had new key chains flying in from Thailand, each capable of holding a single condom. All I needed were 10,000 good prophylactics.

But most manufacturers weren't used to getting orders that large from a single individual. After being turned down by every supplier in the country, I began to have nightmares: I was endlessly in line at Walgreen's buying hundreds of Trojan "family packs." I would wake up wishing that I had followed my classmates into something simple and easy, like investment banking.

Finally, one small-scale manufacturer agreed to sell me the condoms I needed. Fame and fortune were, once again, within reach. Or so I thought.

Then came the fine-print details. According to the Food and Drug Administration, I needed to include a "how to use" guide with each key chain. I realized I needed insurance in case some fool inadvertently Bobbittized himself with my product during a drunken tryst. My conversations with countless insurance agents are among my more forgettable experiences in the business.

"I need product liability insurance," I would say.

"Sure," the agent would reply. "What's your product?"

"Condom key chains," I would answer.

"New market penetration?" the agent would say. This was funny, maybe, the first 10 times.

At long last I had insurance, instruction cards, adhesive logos and 10,000 condoms and key chains. I sat down at the dining room table to put them all together. Two hours later, with my fingers cut and bleeding, fewer than 100 key chains were ready for sale. As I sat there sinking deeper into debt and depression, the only thing I couldn't stop calculating was exactly how big an idiot I was.

Eventually, though, business picked up. People began buying my key chains. Small gift shops in small towns bought dozens and dozens and regularly reordered. I sold thousands to Planned Parenthood clinics—until I received their corporate counsel's certified letter ordering me to "cease and desist" using their "PP" logo or face immediate legal action. I called my friend.

"Congratulations," he said.

"Congratulations?" I said. "What are you talking about?"

"Hey!" he said. "You're not in business until somebody sues you."

Other customers were satisfied to a fault. There was the account in Houston that bought lots of minimum orders for cash. After several months they increased their order tenfold and asked for 30 days' credit. I sent the shipment and never heard from them again. I tried to recover my money and ultimately spent more than they owed me without collecting a dime.

A Bay Area gift shop used a similar technique on me just before Christmas. Desperate for sales, I sent the order. On December 26, the owner simply closed up shop and disappeared.

I learned a few other lessons during my two years as the king of condom key chains. One was that it's tough to get rich quick with a single item. I learned that a markup of 150 percent doesn't mean much when you're only making 75 cents per item. It took a lot of key chains bought at 50 cents and sold for $1.25 just to pay the phone bill. After selling more than

50,000 pieces, I was $10,000 poorer than when I began. When my inventory dropped below 500, I took the remainder to a local advocacy group for prostitutes.

"Do you think you could use these?" I said to the receptionist, who told me her name was Dark Star.

"Sure, baby," she said. "These are hilarious. You gonna give these to us? I bet you could sell thousands of them."

Original article here: https://web.archive.org/web/20140414040320/http://storylog.com/how-my-start-up-failed/

Startup Failure: Wesabe

Why Wesabe Lost to Mint

A number of people have asked and speculated about why the company I co-founded, Wesabe, shut down earlier this summer. Some of the claims or guesses about it are just factually wrong; others seem misinformed to me; others seem to have some truth. I thought I'd add my own opinion.

In November 2006, Wesabe launched as a site to help people manage their personal finances. We certainly weren't the first to try to tackle this problem through a web app, but we were the first of a new wave of companies that came out in the months that followed, characterized but what some would call a Web 2.0 approach to the problem.

Like Flickr and del.icio.us, we relied on community and features such as tags; unlike some of the previous attempts, we tried to automatically aggregate and store all of our users' financial accounts on the web (instead of relying on manual data entry, say, or desktop storage of the data); and most especially, we tried to learn from the accumulated data our users uploaded, and make recommendations for better financial decisions based on that data.

If every copy of Quicken started out as a blank spreadsheet, Wesabe tried to accumulate knowledge from users and data that would fill in some of that spreadsheet for you, and point you in the direction of better choices.

Even before we launched, we heard about other people working on similar ideas, and a slew of companies soon launched in our wake. None of them really seemed to get very far, though, and we were considered the leader in online personal finance until September 2007, when Mint launched at, and won, the first TechCrunch 40 conference. From that point forward we were considered in second place at best, and they overshadowed our site and everyone else's, too. Two years later, Mint was acquired by Intuit, makers of Quicken (and after Mint's launch, the makers of Quicken Online) for $170 million. A bit less than a year later, Wesabe shut down.

That's the history; now, some interpretation, with the completely obvious caveat that I am anything but an unbiased observer. I was the first person to start work on Wesabe and formally co-founded the company (as Chief Product Officer) with my high school friend, Jason Knight (who was CEO). I made nearly all of the product decisions myself, and was notorious with Jason and our board as being very hard-headed about those decisions.

While I relied on many other people in making product choices, I also hired and managed all of those people, so that group was at the least a reflection of who I thought had the right values and ideas. When Jason had to leave the company due to a family illness, I took over as CEO and led the company without a formal peer for the final two years. All that adds up to me having absolutely no one to blame for Wesabe's failure but myself, and as a result I can't now nor could ever be dispassionate in thinking about what happened.

With that in mind, here are what I believe are a number of myths about why Mint won and Wesabe lost:

Mint launched first - I hear this surprisingly often; they didn't. Wesabe launched about 10 months before Mint. More the shame that we didn't capitalize on that early lead. There's a lot to be said for not rushing to market, and learning from the mistakes the first entrants make. Shipping a "minimum viable product" immediately and learning from the market directly makes good sense to me, but engaging with and supporting users is anything but free. Observation can be cheaper. Mint (and some others) did well by seeing where we screwed up, and waiting to launch until they had a better approach.

Wesabe never made any money - untrue. We started generating revenue in late 2008, a year and a half before we shut down; and ran completely out of invested funds almost nine months before closing the company. (Put another way: we survived solely on revenue from mid-November 2009 until the end of July of this year, when we turned off the site.) Obviously we didn't make enough to keep us going indefinitely, but we were not far off from supporting a small company indefinitely on revenue.

Mint was a better name and had a better design - both of these things are true, but I don't believe they were primary causes for our company to fail and for Mint to be acquired. Mint's CEO likes to talk about how ridiculous our name was relative to theirs, but I think the examples of Amazon, Yahoo, eBay, Google, and plenty of others make it plain that even ludicrous names (as all of those were thought to be when the companies launched) can go on to be great brands.

Mint's design, while definitely very appealing and definitely a factor in getting people to trust the company, doesn't seem to me to be enough to explain the different outcomes, again based on what I've seen from other companies (del.icio.us versus Magnolia, eBay versus Amazon Auctions, and now iPhone slipping against Android).

Design matters a huge amount, without question, and Mint's design was exceptional, but if other, stronger forms of lock-in are in place first, design alone can't win a market, nor can it keep a market.

Wesabe wasn't viral and Mint was - half-true: neither product was. Neither of us bore any resemblance to a typical Silicon Valley success story, with traffic surging up and to the right (YouTube, Twitter, etc). Mint aggressively acquired users by paying for search engine marketing (reportedly spending over $1 for each user), while Wesabe spent almost nothing on marketing; yet in the end we grew at about 1/5th the rate they did.

Take a look at Mint's Compete.com graph today and you'll see that their traffic dropped substantially for the six months after their acquisition, and has had sawtooth traffic after that. That pattern matches with a decreased and then increased marketing budget, but not with viral growth. Our patterns similarly followed non-scalable curves (influenced primarily by press wins, economic conditions, and sometimes drafting on Mint's coverage).

Those are what I see as the common beliefs about the outcome of our competition with Mint that I don't believe were right. Why was there such a difference, then?

I see two primary reasons that, if they had changed before Mint's launch or had never occurred in the first place, could well have allowed Wesabe to maintain its early lead and remain the leader against Mint's entry.

First, we chose not to work with Yodlee, but failed to find or make a replacement for them (until too late). Yodlee is a company that provides automatic financial data aggregation as a web service. They screen-scrape bank web sites (that is, read the payee and amount and date by parsing them out of the bank's web site, writing a custom parser for each bank they support). When we talked to Yodlee in 2006, the company was crumbling, having failed to get acquired and losing executives. They were also very aggressive in negotiation, telling us they would give us six months' service nearly free and then tell us the final price we'd be charged going forward. Since they had effectively no competitors, we didn't believe we should tie our company to a single-source provider, especially one in very bad business shape. Mint used Yodlee (at least until they were acquired - I'm not sure what they're doing now) to automatically get user's data from bank sites and import them into Mint, and as a result had a much easier user experience getting bank data imported. Wesabe built our own data acquisition system, first using downloadable client programs (partially because that was easier and partially to preserve users' privacy) and later using a Yodlee-like web interface, but the Yodlee-like version didn't launch until six months after Mint went live, and even then didn't really work as a near-complete replacement for some time after.

A good friend argues that our mistake was not using Yodlee in the first place, and maybe – probably – it was. I believe, though, that we could have made that choice as long as we immediately assumed that someone else would eventually sign up with Yodlee, and that we had to be at least as good if not better than what Yodlee provided, however we got there. (PageOnce, for instance, has not used Yodlee, but has grown very significantly in the same time, using a combination of other aggregation methods that were more effective than ours.) Mint's dependence on Yodlee apparently suppressed their acquisition interest among companies that knew Yodlee well (such as Microsoft, Yahoo, and Google); since we had developed our own technology for aggregation, we didn't have that particular problem, and in fact had some acquisition interest simply for the aggregator we'd built. We just didn't build it nearly fast enough. That one mistake (not using or replacing Yodlee before Mint had a chance to launch on Yodlee) was probably enough to kill Wesabe alone.

Second, Mint focused on making the user do almost no work at all, by automatically editing and categorizing their data, reducing the number of fields in their signup form, and giving them immediate gratification as soon as they possibly could; we completely sucked at all of that. Instead, I prioritized trying to build tools that would eventually help people change their financial behavior for the better, which I believed required people to more closely work with and understand their data. My goals may have been (okay, were) noble, but in the end we didn't help the people I wanted to since the product failed. I was focused on trying to make the usability of editing data as easy and functional as it could be; Mint was focused on making it so you never had to do that at all. Their approach completely kicked our approach's ass. (To be defensive for just a moment, their data accuracy – how well they automatically edited – was really low, and anyone who looked deeply into their data at Mint, especially in the beginning, was shocked at how inaccurate it was. The point, though, is hardly anyone seems to have looked.)

Between the worse data aggregation method and the much higher amount of work Wesabe made you do, it was far easier to have a good experience on Mint, and that good experience came far more quickly. Everything I've mentioned – not being dependent on a single source provider, preserving users' privacy, helping users actually make positive change in their financial lives – all of those things are great, rational reasons to pursue what we pursued. But none of them matter if the product is harder to use, since most people simply won't care enough or get enough benefit from long-term features if a shorter-term alternative is available.

I am, of course, enormously sad that Wesabe lost and the company closed. I don't agree with those who say you should learn from your successes and mostly ignore your failures; nor do I agree with those who obsess over failures for years after (as I have done in the past). I'm hoping that by writing this all out I can offload it from my head and hopefully help inform other people who try to start companies in the future.

You'll hear a lot about why company A won and company B lost in any market, and in my experience, a lot of the theories thrown about – even or especially by the participants – are utter crap. A domain name doesn't win you a market; launching second or fifth or tenth doesn't lose you a market. You can't blame your competitors or your board or the lack of or excess of investment. Focus on what really matters: making users happy with your product as quickly as you can, and helping them as much as you can after that. If you do those better than anyone else out there you'll win.

I think in this case, Mint totally won at the first (making users happy quickly), and we both totally failed at the second (actually helping people). No one, in my view, solved the financial problems of consumers. No one even got close. Yes, both products helped some people – ours mostly through a supportive community and theirs mostly through giving people a rough picture of where their money has gone.

But when we analyzed the benefits we saw for our users, and when Mint boasted about the benefits they saw for their users, the debt reduction and savings increase numbers directly matched the national averages. Because our products existed during a deep financial crisis, consumers everywhere cut back, saved more, and tried to reduce their debt. Neither product had any significant impact beyond what the overall economy led people to do anyways.

So, yeah. Changing people's behavior is really hard. No one in this market succeeded at doing so – there is no Google nor Amazon of personal finance. Can you succeed where we failed? Please do – the problems are absolutely huge and the help consumers have is absolutely abysmal. Learn from the above and go help people (after making them immediately happy, first).

-M

Original article here: http://blog.precipice.org/why-wesabe-lost-to-mint/

Startup Failure: Nouncer

#

Two years is a long time, especially when spent building a startup from the ground up. My most exciting professional adventure to date has reached its conclusion last month when I decided to pull the plug on Nouncer, my attempt at building a microblogging web service. The Nouncer story started with an idea and hunt for logo, evolved into a business with some general directions about a financial model, and materialized into a big chunk of code – 150,509 lines to be exact.

Navigating the Business

Nouncer wasn't an easy idea to explain to people 2 years ago, especially since microblogging didn't exist yet. Twitter, Jaiku, Pownce, and other services were introduced while Nouncer was being developed. The new services made it easier to explain and communicate the idea, but at the same time took away the first-to-market opportunity as well as added competition, and raised questions about the business plan. Ironically, at the same time I left my day job to work on Nouncer full time, Twitter exploded after a huge SXSW exposure.

As Twitter received more attention, clones showed up everywhere. My decision was not to compete with a new and already crowded space, but to find a different angle. I decided to focus on scalable technology, making a bet that once Twitter and others will get popular, there will be a real need for that kind of technology by other players.

While forgoing the plan to build a consumer facing application, I still planned to focus on the enterprise world where I spent my 10 years exile from the web community. I still believe that microblogging is going to mature and change the way corporations communicate internally and with clients. As a Vice President at Citigroup managing a team, I was forced to read over 500 emails a day, spending at least 2 hours every morning just catching up. My plan was to start working on a product for the enterprise as soon as the backend framework was complete.

The business decision to focus on technology and avoid building a consumer application had a significant impact. It meant throwing away 3 months worth of consultant work building a Perl-based website that interfaced with the backend database and engines, and provided a user interface. But the most significant change, which I did not appreciate until it was too late, was the fundamental shift in the nature of the startup. I moved from building a consumer-facing website to a combination of infrastructure provider and software developer which made the overhead cost much higher, time to market longer, and resources much more difficult to find.

A month ago, half way through my angel funds raised from family members, I decided to review the progress I've made and figure out what still needs to happen to make this a viable business. I was also actively pursuing raising VC funds with the help of a very talented and well connected friend. At the end, I asked myself what are the most critical resources I need to be successful and the answer was partners and developers. I've been looking for both for about a year and was unable to find the right people. I realized that money was not the issue.

To me, that implied that raising funds is not going to solve my problem. While it might make the startup more attractive for others to join, I failed at building a team. I also approached people I know and respect with a what-if scenario. I asked them if I raised $1M, will they join me. I got a wide range of 'no' reasons and was still at square one. There are plenty good reason why people would not leave a job, especially a high paying financial services job where most of my friends work. This is New York after all.

With half my angel funds in the bank, and a couple of really promising VC meetings, I made a decision that surprised most people, and pulled the plug. This was actually an easy decision because the fundamental facts were easy to understand. I had no reason to believe I'll do a better job finding a team with more money, and I couldn't have moved fast enough without a team. So I gave the money left back to my family who supported me (which after some tax planning will hopefully turn into a very small loss), and explained why I thought it was the better move for me and them.

Managing Distractions

Looking back, it is amazing how much impact small decisions have. Halfway through the project, right around the introduction of the Facebook platform, the work on Nouncer produced an interesting side project. It wasn't a perfect fit for the Nouncer services, but still fit in with the overall strategy and philosophy. It also looked like an easy thing to do with a big marketing potential. The result was JabAbout, a Facebook application using the social graph to propagate short messages by following the friends-of-friends paths. JabAbout failed to build a user base and was eventually shutdown.

The financial and schedule impact JabAbout created wasn't by itself dramatic, which doesn't mean to say it wasn't a big waste of time and money. But it happened at the same time I was reshaping the product and the business strategy, and the two efforts combined, while still coding Nouncer, was too much distraction. The lesson, of course, is not to get distracted trying to accomplish small wins. When the Facebook platform was announced, it was such a significant event, that not coming out with some application seemed to create a disadvantage.

This brings me back to the underlying problem – I didn't have a partner to balance me out and provide sanity checks for business and technology decisions made. And there were plenty of those to review and debate. I didn't start solo. A few months after I came up with the initial idea for Nouncer, I recruited two good friends to join me. The three of us formed a partnership with a simple agreement and divided the work between us. We were all fully employed at the time and could only dedicate a limited amount of time to the startup.

As is common in such cases, not everyone was able to contribute at the same level. My personal motivation for putting in hours of work late at night every day after coming back from the office was, well, not to go back to the office. I was trying to create a viable alternative to my day job. My friends and partners did not share my sentiment, at least not at the same level and after a month of discussions, decided to dissolve the partnership. We were extremely fortunate to be able to accomplish that with no impact to our friendship, and I retained full ownership and control over the code and other intellectual property created together.

The decision to move on alone wasn't by itself wrong. I've set milestones and committed myself to finding new partners and completing parts of the system by certain dates. As is often the case in projects with no risk management or review process, every single milestone was either missed or changed. This did not result from lack of project management experience or understanding as I am actually a certified project manager, but was the result of having no one to counter my decisions, questions them, and provide feedback. My ability to talk myself into anything was playing against me.

Selecting Technology

When working on the initial concept behind Nouncer, what today we simply call microblogging, I immediately identified the need for massive scalability. In early conversation I was worried about overnight success shutting the service down for months while we tried to build it again to sustain the load. Messaging systems are not trivial and require a big investment in infrastructure, both software and hardware. This, together with my decade-long experience building and architecting high-volume high-frequency trading systems, led me to choose C++ as the language of choice.

The technology decision to use C++ was well founded. However, it failed to take into account, or at least prepare for the inevitable reality of being unable to find people to join me. Any major web company reaching massive scale eventually turns to C/C++ for performance. Given the fact that Nouncer had no competition at all when the idea was first conceived (and I am not claiming to have invented microblogging – only that I came up with the idea independently), I decided to take a little bit longer before launching to build a solid foundation. This, of course, turned up as a big mistake – but it is far from clear that I would have been first to market either way.

Many people criticize the typical path Web 2.0 applications take in their development: putting together a poorly executed site, gauging the market, and only upon success building the service to actually scale and accommodate the market. However, the cost of building scalability ahead of time is extremely high, and for most startup is cost prohibitive. The decision to turn Nouncer from a consumer service to an infrastructure provider forced a significant early investment that had very little change of materializing. It is the kind of project entrepreneurs get to work on once they have established their reputation and funding connections.

A close friend reading my early business presentation told me he liked it a lot but was worried about one line. He said it sounded like I'm building infrastructure and that is going to create problems with most investors. At the time I was focused on building a website and the comment didn't register. Now with more perspective I can certainly appreciate the advice. While I still stand behind my technical decision to choose C++, it was the wrong business decision. I should have started

with a dynamic language with wide community support and moved components to C++ as needed.

Figuring Out the Product

In my head, Nouncer always had a clear vision. Funny how that is not enough to get other people excited about your project. The greatest benefit of turning Nouncer from a consumer website to a backend API service was that I no longer needed to hire a great user interface designer or developer. The product was now a set of API calls and a document explaining how to use them. Amazon proved beyond any doubt that this is a valid and powerful business model, but very few companies can afford to build such a service and then wait for people to come. It also helps that Amazon did not start with the goal of providing these services, but identified an opportunity once their own retail infrastructure was in place.

When I finished the first alpha release of Nouncer, I was eager to get people to try it out and play with it. However, the lack of front end turned out to be a major obstacle. Most people need a working piece of code as a starting point and since all I had was a list of API calls, the learning curve was too steep and the product failed to inspire and excite its target audience.

Talking about Nouncer I was constantly shifting the product message focus. I made little changes to the software roadmap throughout the project, and had a pretty solid understanding of what needs to be built to support all the various plans I had for the product. But when talking about it, I focused on the particular implementation or use case I was most excited about that day. One day it was enterprise software, another it was helping small retail stores reach their local community, and sometimes it was a tool to enhance existing social networks. All of these use cases required the exact same framework, but not sticking with a single message created a lot of confusion about the product. To this day most people I talked to don't really understand what I was building.

Dealing with Location

My fellow NextNYers are all too familiar with the constant debate if New York is the right place to start a web company. Like most people, I decided to start Nouncer in the New York area because this is where I live. It is one thing to relocate a family while maintaining job security, but a whole different matter when considering relocation with the uncertainty of startup life. It also took some time, almost a year, to appreciate the differences between Silicon Valley and Silicon Alley, and to realize that the kind of startup I wanted to build was not being enabled by my environment. This doesn't mean you can't build successful web companies in New York, a statement I found fundamentally wrong.

Very few technology companies can compete with Wall Street when it comes to offering cash to developers. The reality of the job market in New York dictates a higher cost for many technical and creative roles. There is also something to be said about the lack of startup culture which is all too common in the Bay Area. When people look for a job in New York, they look for established players where they can build their reputation and business network.

In my case, New York didn't lack money, community, able workers, or smart people with good advice. It lacked the audience my product needed to succeed – the early adopters web hackers looking for the next cool toy to play with. I can count on one hand the number of people I've met in New York meetups and events that fit this description. In San Francisco one can find hacking events on a daily basis, as well as many unconference events where people get their hands dirty playing with code. This is a very specific audience dictated by my decision not to build a consumer product.

One of the early decisions I've made building Nouncer was to use as many open standards as possible. I found microformats and OpenID extremely appealing and wanted to incorporate them into the platform which introduced me to the communities working on these and other standards. There are plenty of companies working in this area making significant contribution to web standards, but for the most part, these are done as part of large standard organizations. The work I was interested in was all community-driven, and that community was mostly centered in the Bay Area.

There is nothing wrong with starting a tech startup in New York. In fact, it is one of the best places to do so. People constantly complain about the lack of talent, the inability to compete with banks, the non-existent "hardcore" tech community, and other nonsense. I used to do it myself. It is so much easier to blame the city than take a deep look at what you are doing and realize that not every business fits every locality. After all, the point of a business is to sell products, and if you the target market is elsewhere, the best talent and resources are not going to make a difference. It all boils down to understanding your business.

It Wasn't All Bad

Actually, it was all good. Most startups fail – that is a fact. But while Nouncer failed to deliver a product and lost some money, it was an amazing adventure. I was fortunate to have the financial stability and family safety-net to do this for this long. There were many sacrifices from a much lower spending ability to family stress and uncertainty. But life working for my former employers had other problems, most importantly that I wasn't happy. Nouncer also have the distinction of stopping when it made sense, not when it ran out of cash.

Working on Nouncer created my need for standards that were not readily available, a fact that got me involved in OAuth and joined the open standards community I had no chance of entering without, as well as the local entrepreneurs community. Rebuilding my professional network outside the financial services industry would not have been as successful or even possible without the context of attempting to build a startup. Two years ago I was practically unemployed outside the financial services industry which was more than happy to offer me the exact same role over and over again.

There are many lessons learned from this experience. I cannot point to any single mistake or error in judgment as the most important one, but I can point to a single positive lesson learned without hesitation – participation and contribution to the community, local and virtual alike. Working within a community yields more valuable return than any other investment. It is also what will eventually protect you more than anything else if your startup fails. The lack of business success will pale in comparison with the relationship and reputation you build.

My next adventure is going to be even more exciting, and would not have been possible without Nouncer. I would like to thank everyone who offered advice, listened to my pitches, reviewed documents, was there for me when I needed help or just someone to bounce ideas off. I wanted to list the names of people who's help meant the world to me but the list is just too long, and that by itself fills my heart with happiness.

I want to thank everyone from the OAuth and NextNY communities – my two virtual homes. It is hard to explain just how much I have learned and enjoyed being part of these two amazing communities (and hope to continue). But most importantly, I want to thank my family for supporting me, believing in me, just being there for me, and suffering through it all with me when things were not so clear.

As for what's next, well, stay tuned…

Original article here: http://hueniverse.com/2008/04/25/the-last-announcerment/

Startup Failure: BricaBox

BRICABOX: GOODBYE WORLD! /

Today, I'm announcing my plans to close BricaBox, LLC.

This decision, which has taken place over the last few weeks, was as complicated as are my feelings about it. Nonetheless, I can tell you most of this decision revolved around issues of money, traction, team, and vision: the four essentials of a successful startup.

I think it's fair to say that a startup deserves to live if it has good quantities of at least three of those four things, and BricaBox is now out of all but one of them.

Perhaps, even, it was out of them sometime ago -- this is where that quote about Hillary Clinton comes in -- and we've just been in denial.

Whatever it is, I'm going to take a tremendous amount of experience, lessons, wisdom, etc with me. And, over the next few weeks, I intend on blogging diligently about every aspect of this failure. I've taken extensive mental notes on these lessons, and I look forward to sharing them with you. I think this process will help me institutionalize these lessons for myself, and of course I hope you can learn something from them as well.

For now, I'll give you a table of contents for what's to come. Subscribe to my RSS feed to be alerted to when I post and see below for some teasers.

As for what I intend on doing next, you can be sure that it will be something entrepreneurial. A few weeks ago, I spoke to a few of the companies I admire most in New York (about joining their ranks), but realized quickly that I was not ready to take my hand off the entrepreneur's tiller.

So, I plan on doing a mix of things: consulting for media and technology companies, launching some exciting small projects with my friends (soon to be announced), and exploring new startup opportunities.

Of course, with all these things, I now have the perspective of building and launching BricaBox -- and I'm excited to put those lessons to work. Also, I bring with me my entire life experience, dating back to my first company (Westheimer Family Plants and Produce), building and running Brandeis Television, directing new media operations at National Public Media, and working with political organizations like David Pepper's campaign for Mayor and now TruthThroughAction.org's project.

I've realized that I'm a "new media mechanic" -- one part technical, one part zen -- so, if I can help you tune-up your online media operations, please let me know.

As for what I intend on doing with BricaBox itself, time will also tell. We'll keep the site live for now. However, if there's a media company in need of a proprietary platform to scale thousands of user-generated content websites, I think BricaBox could be of great use for them. Of course Open Sourcing is another option, though that would also require a certain level of interest from the community.

Last of all, I want to thank several people who made the experience with BricaBox wonderful (though there were countless who had a great effect on the experience): Thanks to the nextNY community for free business school and for the friendships; thanks to Evan Bartlett, Michael Galpert, Charlie O'Donnell (my biggest -- always constructive -- critic), Alex Lines, and to David S. Rose, for his mentorship. A big thanks goes to my dad for his watchful eye. Many thanks go to Gary Vaynerchuk for his commitment to my success. To Allen Stern of CenterNetworks, the great guys at Silicon Alley Insider and Caroline McCarthy of CNET, thanks for promoting New York Tech companies!

Thanks to Mike Hostetler for stepping into the technical lead when we were most in need.

And lastly, thanks to everyone who loved BricaBox like I did. We had some awesome users and fans, like Nichelle Stephens, Andrew Watson, Ari Greenberg, and more. Asking them to say goodbye to BricaBox hurts the most!

As I write the final words of this post, my mind is still coming up with reasons and ways not to do this; alas, it must be done -- it is that time. Now, I get to look onwards and upwards. As I look at the table of contents (below) to my startup's postmortem, I get excited about working on what's next. The opportunities I have coming are incredibly exciting, and applying all that I've learned to the next Big Thing will make it all worth it. That's for sure.

POSTMORTEM TABLE OF CONTENTS (COMPLETE POSTS TO COME)

Companies should tackle Market Problems, not Technical Problems. BricaBox was a solution to a technical problem I had. While it's good to scratch itches, it's best to scratch those you share with the greater market. If you want to solve a technical problem, get a group together and do it as open source.

Start with a real team. There are a million things a startup needs to do and a dozen skill sets. If you get more people involved from the get-go, you can better distribute responsibility, and grow on the cheap.

Lightweight or heavyweight? Choose one. It matters how you spend. Does your startup have a burn-rate? If it's above $0, think about concentrating it and speeding up development. BricaBox was fed-but-anemic and and slowly roasting its cash. Do it again and I'll concentrate those costs at the beginning. We would have had twice the product in 2/3 of the time.

When in doubt, build off Open Source. One of the first questions I had to deal with, while building BricaBox, was why we weren't modifying an existing Open Source solution, like Wordpress MU. We were a CMS at heart, after all. Next time, I'll give more consideration to building off and participating in existing Open Source project.

Go vest yourself. When a co-founder walks out of a company -- as was the case for me -- you've already been dealt a heavy blow. Don't exacerbate the issue by needing to figure out how to deal with a large equity deadweight on your hands (investors won't like that the #2 stakeholder is absent, even estranged, from your company). So, the best way of dealing with this issue is to take a long, long vesting period for all major sweat equity founders.

Original article here: http://innonate.com/2008/06/19/bricabox-goodbye-world/

Startup Failure: YouCastr

YouCastr - A Post-Mortem

Introduction: Why I'm writing this

There are many post-mortems from failed startups out there, mainly because there are a lot of failed startups, and the people that start them tend to be very introspective and public about their successes and failures. I'm no different. This post-mortem will serve to get things off my chest, organize my thoughts, get the most out of the experience, and share my experience with others.

I'm also writing this to be able to point to a single, detailed, lengthy answer to the inevitable questions I'll be getting from friends and colleagues about what happened with YouCastr. Now people can read to their heart's content.

As I write about my experiences, people who know startups will see many common themes. Most of the conclusions here are consistent with Paul Graham's analysis and understanding of startups, what's challenging, and why they fail (if you're planning on starting a company, especially in tech, read every single one of his essays). But no matter how many times you read or hear certain lessons, you have to do it yourself to really understand. Here are my specific and personal experiences to add color to that advice.

A Brief History

YouCastr is Born

The idea for YouCastr was hatched about three and a half years ago, during a car ride down from Jay Peak with my good friend Jeff Dwyer. Throughout the entire ride we were throwing ideas back and forth about cool companies, fun ideas, and stuff that bothered us. One in particular stuck with me, which was the idea of bringing Mystery Science Theater 3000 to the modern age, and let anyone provide their own comments. After exploring this some more, I ended up settling on sports as the logical market. This would mean the commentary would have to be live, because otherwise it would have been just a podcast platform, which was not very interesting technologically. A couple of days later I shared the idea with my soon to be co-founder Jeff Hebert, and we continued thinking about it. Within a few months we put together a team of 4, created our initial plans, and began building an alpha version.

Initial progress

As we were getting off the ground, three of the four founders (including myself) were mainly working on building the actual product, since we couldn't do much of anything without that. We learned Ruby on Rails, did our own designing, worked nights and weekend on the startup as a complement to our management consulting day-jobs, and clawed our way to our initial alpha and private beta.

Building the company

We then spent the next three years building the company, quitting our jobs, raising money, opening our office, hiring people, firing people, going back to bootstrap mode, and finally, pulling the plug.

Pivoting

Startups are all about pivoting. It's like a word game where you start with HURT, change one letter at a time, then end with LOVE, with many other four-letter words along the way (btw, you really can do that in 6 moves). In our case, we started as a virtual sports bar where people could chime in audio commentary and ended up as a do-it-yourself pay-per-view video platform. Seems like a radical change, but each pivot was the equivalent of changing one letter.

Our first pivot was to focus on live audio sports broadcasting of games that weren't being televised (instead of adding commentary to nationally televised games), in response to what customers were doing. Then we added video broadcasting to that same market, focusing on broadcasting sports that weren't covered (primarily high school and college). And finally we expanded beyond sports, mainly by de-emphasizing the sports branding on our platform, and adding a few features more geared towards video producers than to schools and teams. All of these were natural pivots,

not jumps.

Shutting Down - The Reasons and Need

1 - Ran out of cash

The single biggest reason we are closing down (a common one) is running out of cash. Despite putting the company in an EXTREMELY lean position, generating revenue, and holding out as long as we could, we didn't have the cash to keep going. The next few reasons shed more light as to why we chose to shut down instead of finding more cash.

2 - The market was not there

The thesis of our current business model (startups are all about testing theses) was that there was a need for video producers and content owners to make money from their videos, and that they could do that by charging their audience. We found both sides of that equation didn't really work. I validated this in my conversations with companies with more market reach than us, that had tried similar products (ppv video platform), but pulled the plug because they didn't see the demand for it.

Video producers are afraid of charging for content, because they don't think people will pay. And they're largely right. Consumers still don't like paying for stuff, period. We did find some specific industry verticals where the model works (some high schools, some boxing and mixed martial arts events, some exclusive conferences), but not enough to warrant a large market and an independent company.

3 - The team was ready to move on

The core team of 5 of us has already made significant personal, financial, and emotional investment over the past three years. We have had our share of tough breaks, not to mention almost two years without salary.

4 - No light at the end of the tunnel

Given the industry trends we were seeing, we just didn't see a light at the end of the tunnel worth surviving for. This market may get slightly better, but it's not going to be a big company.

5 - We need to put a stake in the ground

We're survivors. We hate giving up, and have tenaciously survived for a long time. We survived the worst economic environment in generations (during which we also tried to raise money), and knew how not to die. But it was time to shut down and move on.

A Retrospective - Why we failed

In most of the following points I say we, but as CEO, many of the mistakes below can be placed directly on my shoulders, which I will take as a valuable experience as I move on.

Didn't pivot fast enough

We did a pretty good job of pivoting, but we didn't make the hard choices quickly enough. Our first pivot, towards broadcasting of original sports content, took us 6 months. We had legacy users and web traffic that we were afraid of losing. We had a brand in one area. We had a story for investors, customers, and the community. All of that made it harder for us to quickly change our focus and apply all resources to the new area, which we knew was the long term future. We iterated our product quickly, but didn't pivot fast enough.

Didn't love it We started the company because we liked the idea and wanted to do something entrepreneurial. We weren't in love with the idea or market we were going after, and weren't core users of our product. We worked really hard getting it off the ground despite this, but it made it more difficult to sustain the energy and to understand the best product choices.

Made hiring mistakes

Hire slow, fire fast. That's how the saying goes. We made a mistake when hiring. We hired too early (before product-market fit), and we rushed into it by foolishly setting an internal deadline to make the hiring decision, which drove us to ignore some gut reactions by some of the core team members. Eric Ries has a great overview of how not to make these mistakes in his Lean Hiring Tips post.

Spent too much time raising money

Almost immediately after our private beta we started trying to raise money, and saw many of the typical challenges when raising money for the first time. This was late 2007 remember, when plenty of Web 2.0 companies were getting funded, and we got caught up in the spirit. We spent 6 months raising our angel round, and finally closed it after launching our public beta in February 2008. We raised less than we should have (Chris Dixon notes that the worst thing a seed-stage company can do is raise too little money and only reach part way to a milestone, which is great advice). We spent 3 months getting ready for raising our next round, then 6 months beating our head against a wall trying to raise money during the financial collapse. Finally we regrouped, but had serious short term cash issues that we had to resolve with additional investment from current investors and cutting of costs. We then focused on the product and made some solid progress, with the plan on giving it another go. By the time we got there, though, we realized we didn't have the metrics we wanted, so pulled back our fundraising efforts and decided to go back to bootstrap mode.

Never quite figured out customer acquisition

We never seriously figured out customer acquisition, and had trouble growing throughout our various pivots. Customer acquisition is hard and more expensive than most people realize when starting a company. There has been a lot of good analysis, including Andrew Chen's overview of calculating cost per customer acquisition and David Skok's warning about how it can be a startup killer, but it takes firsthand experience to understand that.

Too many founders

We started the company with four co-founders. I believe Dharmesh Shah nailed it in his analysis of the optimal number of co-founders. In our case, the advantage of having 4 co-founders was that had more people and resources willing to work for sweat equity. But it also presented challenges, including slowing down decisions, as everyone naturally wants to chime in with their opinion. As CEO, a lot of this falls on me, as I should have been stronger in making decisions. I tend to be a consensus builder, but in my next startup I will have to improve my ability to make bold unpopular decisions while still motivating everyone and building consensus after the decision is made. On the plus side, the entire founding team was incredibly committed through the entire time, and we never had any of the fundamental founder issues that many people complain about become a major problem.

Counted on big partnerships to close (when they didn't) and to help more than they did

We initiated some very interesting high level partnership and business development discussions with large companies. We put some hope into these deals, but large companies move much slower than startups, and every deal took much longer than we needed. Marc Andreesen even draws a parallel that going after large partnerships is like chasing Moby Dick.

We put too much stock and hope into closing some of these. And for the ones we did close, we expected more results than we got. It's an interesting dilemma, because partners are often the best acquirers, but the downfalls can even lead people to suggest to NOT partner with big and powerful companies.

What I Learned and some of the positives

I'm an optimist by nature, which you have to be to start a company given the odds and challenges. Given that, it's good to look back and think about the positives of the entire experience. It's incredibly satisfying to create a revenue-generating company that is flirting with cash-flow positive, especially having started with a crazy idea. The experience of building a company and tasting success was incredible, and though we didn't quite get over the hump, we have tasted it, and it will only motivate me more to get there again, through all the hardships.

On a personal level, I've learned more about business, people, life, and myself over the past three years than I ever have. I have made life-long friends with whom I can share these experiences with. I have come closer to understanding how to reach my limits, which is one of my fundamentals goals in life. Startups let you try and find those limits in a way that most

jobs wouldn't, and I'll keep pushing. And best of all, I have realized that my true passion is entrepreneurship. I have a very addictive personality, and therefore only do things where I am comfortable with the extreme case, like starting companies or running ultra-marathons.

We built a great culture, including some great parties for the local tech-scene, competitive ping-pong matches, and late night coding sessions with house music blasting. We converted three lifelong PC people to Mac, all of whom are happier for it. And most satisfying, we helped kick start two young and promising careers, those of our designer and community manager.

Moving Forward

My long term goal is to continue starting companies. There's no question I'm in a better position now to start another company than I was when I first started YouCastr. It's almost like thinking back at how much more fun high school would have been had you known what you knew when you graduated back when you started. Practically speaking, though, after having gone almost 2 years without a salary, I'm not in a financial position to bootstrap another company, which is the way I would really want to start a company.

In the short term I'm working on various projects for my consulting company, Scenario4. We are working with some great people on some fun projects across the tech world. I am also pursuing several side projects, including the recently launched BooksforBits, a non-profit company designed to help accelerate the transition to ebooks while getting books to communities in need.

It's an incredibly exciting time in technology. We are on the cusp of the post-pc era, which includes a revolution in the mobile space, and in how we interact with and perceive computers. We are in a world where startups can be more capital efficient than ever before. And most interestingly for me, as a society we are going to be facing challenges unlike any we have ever seen.

With every ending is a new beginning, and I'm excited about what lies ahead.

Original article here: http://theambitiouslife.com/post/48691070654/youcastr-a-post-mortem

Startup Failure: Devver

Lessons Learned

As we've begun to wrap things up here at Devver, we've had the chance to reflect a bit on our experience. Although shutting down is not the outcome we wanted, it's clear to both Dan and I that doing a startup has been an amazing learning experience. While we still have a lot to learn, we wanted to share some of the most important lessons we've learned during this time.

The community

When we started Devver, we were hesitant to ask for feedback and help. We quickly found that people are incredibly helpful and generous with their time. Users were willing to take a chance and use our products while giving us valuable feedback. Fellow Rubyists gave us ideas and helped us with technical problems. Mentors made time for meetings and introduced us to others who could assist us. And other entrepreneurs, both new and seasoned, were happy to share stories, compare experiences, and offer support.

If you are working on a startup, don't be afraid to ask for help! The vast majority of people want to help you succeed, provided that you respect them and their time. That means you need to prepare adequately (do your research and ask good questions), figure out their preferred methods of communication (e.g. don't call if they prefer email), show up on time, don't overburden them, and thank them. And when other people need your help, give back!

You can build awesome relationships with various communities on your own, but we strongly recommend joining a mentorship program like TechStars. The program accelerated the process of connecting with mentors, users, and other entrepreneurs by providing an amazing community during the summer (and to this day). The advice, introductions, and support have been simply incredible.

Founding team

Dan and I are both technical founders. Looking back, it would have been to our advantage to have a third founder who really loved the business aspect of running a startup.

There is a belief (among technical founders) that technical founders are sufficient for a successful startup. Or, put more harshly, that "you can teach a hacker business, but you can't teach a businessman how to hack". I don't want to argue whether that's true or not. Clearly there are examples of technical founders being sufficient to get a company going, but my point is that having solely technical founders is non-optimal. You can teach a hacker business, but you can't make him or her get excited about it, which means it may not get the time or attention it deserves.

Hackers are passionate about, well, hacking. And so we tend to measure progress in terms of features completed or lines of code written. Clearly, code needs to be written, but ideally a startup would have a founder who is working on important non-technical tasks: talking with customers, measuring key metrics, developing distribution channels, etc. I'm not advocating that only one founder works on these tasks while technical founders ignore customer development – everyone needs to get involved. Rather, I'm pointing out that given a choice, technical founders will tend to solve problems technically and having a founder who has the opposite default is valuable.

Remote teams

We embraced working remotely: we hired Avdi to work in Pennsylvania while Dan and I lived in Boulder and later on, Dan moved to Washington, DC. There are many benefits to having a distributed team, but two stood out in our experience. First, we could hire top talent without having to worry about location (in fact, our flexibility regarding location was very attractive to most candidates we interviewed). Secondly, being in different locations allowed every team member to work with minimal distractions, which is invaluable when it comes to efficiently writing good code.

That said, communication was a challenge. To ensure we were all synced up, we had a daily standup as well as a weekly review. When Dan moved to DC, he and I scheduled another weekly meeting with no set agenda to just bring up all the issues, large and small, that were on our minds. We also all got together in the same location every few months to

work in the same room and rekindle our team energy.

Also, pair programming was difficult to do remotely and we never came up with a great solution. As a result, we spent less than a day pairing a week on average.

The most significant drawback to a remote team is the administrative hassle. It's a pain to manage payroll, unemployment, insurance, etc in one state. It's a freaking nightmare to manage in three states (well, two states and a district), even though we paid a payroll service to take care of it. Apparently, once your startup gets larger, there are companies that will manage this with minimal hassle, but for a small team, it was a major annoyance and distraction.

Product development

Most of the mistakes we made developing our test accelerator and, later, Caliper boiled down to one thing: we should have focused more on customer development and finding a minimum viable product (MVP).

The first thing we worked on was our Ruby test accelerator. At the time, we thought we had found our MVP: we had made encouraging technical progress and we had talked to several potential customers who were excited about the product we were building. Anything simpler seems "too simple" to be interesting.

Our mistake at that point was to go "heads down" and focus on building the accelerator while minimizing our contact with users and customers (after all, we knew how great it was and time spent talking to customers was time we could be hacking!). We should have asking, "Is there an even simpler version of this product that we can deliver sooner to learn more about pricing, market size, and technical challenges?"

If we had done so, we would have discovered:

whether the need was great enough (and if the solution was good enough) to convince people to open their wallets that while a few users acutely felt the pain of slow tests, most didn't care about acceleration. However, many of those users did want a "simpler" application – non-accelerated Ruby cloud testing.

the primary technical challenge was not accelerating tests, it was configuring servers for customers' Rails applications. Not only did we spend time focusing on the wrong technical challenges, we also made architectural decisions that actually made it harder to solve this core problem.

After eventually discovering that setup and configuration was our primary adoption problem (and after trying and failing to implement various strategies to make it simple and easy), we tried to move to the other end of the spectrum. Caliper was designed to provide value with zero setup or configuration – users just provided a link to source code and instantly got valuable data.

Unfortunately, we again made the mistake of focusing on engineering first and customer development second. We released our first version to some moderate success and then proceeded to continue to churn out features without really understanding customer needs. Only later on, after finally engaging potential customers did we realize that market was too small and price point was to low to have Caliper sustain our company by itself.

Conclusion

This is by no means a comprehensive list, but it is our hope that other startups and founders-to-be can learn from our experiences, both mistakes and successes. Doing a startup has been an incredible learning experience for both Dan and I and we look forward to learning more in the future – both first-hand and from the amazing group of entrepreneurs and hackers that we've been privileged enough to know.

Original article here: http://devver.wordpress.com/2010/04/26/lessons-learned/

Startup Failure: Overto

Lessons Learned: Startup Failure

Some time ago we closed down Overto – startup I was involved in. It was a failure – pretty obvious thing since we've closed the service. Since we learn much on our mistakes I think a reliable analysis why the business have failed should be valuable for you. For the beginning things we screwed.

No one working full time

From the very beginning we knew none of people engaged is willing to leave their daily jobs to commit fully to the startup. We thought we'd able to run internet service after hours. To some point that was true. As far as nothing bad was happening with the servers and the application it was all fine. We were working on new features when we had enough free time. Problems started when we faced some issues with our infrastructure. We weren't able to resolve issues on the fly and had several downtimes. You can guess how it influenced user experience. That also backfired on service development since we had to focus on current problems instead of adding new functionalities. Lack of person working full-time and being able to deal with maintenance and bug fixing was the most important reason of failure.

Catching the market

That's partially a consequence of the previous point. Since we spent majority of our limited time on trying to keep the service running we weren't able to catch the changing market. We needed to cover new areas to move the application to another level but we couldn't complete ongoing development. The whole project stopped in beta version and for users it didn't look like anything was about to change.

Lack of marketing skills

Thin line between life and death of internet service is a number of users. For the initial period of time the numbers were growing systematically. Then we hit the ceiling of what we could achieve effortlessly. It was a time to do some marketing. Unfortunately no one of us was skilled in that area. Even worse, no one had enough time to fill the gap. That would be another stopper if we dealt with the problems mentioned above.

Business model

We hadn't checked very well a business model we set up on the beginning. We were surprised a couple of our features weren't as unique as we'd initially thought. Ironically that wasn't a big problem since we had a bunch of ideas how to adjust the strategy in a new situation. Anyway, you should plan to change your initial business model.

Screwed chance of selling the business

After we'd decided we won't be able to maintain the service in the long run we had a chance to sell it. To make a long story short we screwed negotiations starting with way too high price. We thought more about how much work we put into the project than how much it can be worth for potential buyers. Things are worth as much as one's willing to pay for them, no matter how long it took you to produce them.

Waiting too long with final decisions

I think that one is a bit sentimental. Since the service was our child we were reluctant to make a decision about closing it faster and limit losses. We've been tricking ourselves thinking that everything would be fine while we couldn't get the application back to work properly.

Setting up a company behind

Setting a company, which is quite an effort in Poland, from the very beginning was really a good idea. All founders knew each other well so lack of trust wasn't the main reason standing behind the decision. We just wanted to give ourselves a

bit of motivation making it a real business. However thing we didn't really plan was that we gained much credibility showing company's details instead of letting people think the whole thing was created by some student during holidays and won't be supported in any way in future. Seeing a company behind a service doesn't guarantee you it won't die but at least you can be sure someone cares.

Long discussions before start

Before launching the project we had very long discussions about what we plan to do and how we want to do it. Of course not every detail was checked but when I compare Overto to other startupish projects I was working on it was really well-thought at the beginning.

Wide range of roles covered with our experience

We had really good pack to run an internet service. Development, administration, user support – in all those areas we had experience from the past. There weren't many things which could have surprise us. We weren't forced to look for a specialist in any technical aspect of building our application.

Working in the same place

For some time we worked in the same building which helped much in decision-making process. We could all meet ad-hoc whenever something important had to be decided. After some time we spread among different places and suddenly we saw how much value was in working in the same office.

Despite we invested some money to fund the project I don't treat it as a loss. For experience I gained it was a low price. Another time when I'll decide to jump to that kind of project chances of success will be bigger.

Original article here: http://brodzinski.com/2009/01/lessons-learned-startup-failure-part-1.html

Startup Failure: Monitor110

Monitor110: A Post Mortem

Turning Failure into Learning

Writing a post mortem is hard, particularly when the result is failure: a failed deal; a failed investment; a failed concept. That said, without a post mortem, without deep reflection, honesty and introspection, how can we get better and do better the next time? Quite simply, we can't. My involvement with Monitor110, as an investor, Board member and leader, was one of the most interesting and informative experiences of my life. I learned about areas I never dreamed of. I worked with a terrific group of young, exceptionally bright people who believed in the vision. Ultimately, we failed. But why did we fail, and what could we have done differently? Some of the stuff is pure 20:20 hindsight. These observations aren't worth much. But the interpersonal dynamics, the issues of organizational structure, the need to change strategy in light of new information, the relationship with key investors, all of these are very instructive. I will endeavor to be as honest and candid as possible.

Let me say that I deeply respect everybody involved with Monitor110 from the original founder, Jeff Stewart, to our investors, employees and customers. Everyone tried very hard to make things work, and this post is not an indictment of anyone. There are no "bad actors" in this story. But a confluence of factors made success an uphill struggle.

The Seven Deadly Sins

While we certainly made more than seven mistakes during the nearly four-year life of Monitor110, I think these top the list.

- The lack of a single, "the buck stops here" leader until too late in the game
- No separation between the technology organization and the product organization
- Too much PR, too early
- Too much money
- Not close enough to the customer
- Slow to adapt to market reality
- Disagreement on strategy both within the Company and with the Board

A Little Background

I was initially approached in early 2005 by Jeff Stewart, who had the original idea for Monitor110. It was a compelling idea. The thesis: more and better information is being put out on the Internet every day, information that can be valuable to Institutional investors who are constantly looking for an edge. And these investors were not very sophisticated about how to best access this information; Monitor110 would use technology to help them get that edge.

Jeff and a few guys had hacked together a version 1.0 of the system, which was based on a boolean matching engine with rules corresponding to each company and investment theme. It was fast. It worked ok. We spent some time working with PubSub, who had built a scalable matching engine but was not focused on the financial services industry.

By mid-2005 the system worked, but spam was becoming more prevalent and caused the matching results to deteriorate, e.g., too much junk clogging the output. Around the same time we started to dig into natural language processing and the statistical processing of text, thinking that this might be a better way to address the spam issue and to get more targeted, relevant results.

This prompted us to not push version 1.0, instead wanting to see if we could come up with a more powerful release using NLP to mark the kick-off. In retrospect, this was a big mistake. Mistake #5, to be precise. We should have gotten it out there, been kicked in the head by tough customers, and iterated like crazy to address their needs. Woulda, shoulda, coulda. Didn't.

An Unusual Leadership Structure

The idea was that Jeff was the technology guy and I was the business guy. Jeff focused on technology and product and I

focused on fund raising, HR, controls and client access (given my Wall Street and hedge fund rolodex). On paper, made sense. Jeff was a successful three-time entrepreneur, I was an experienced senior Wall Street executive.

The problem was, however, that when it came time to make hard decisions the two-headed structure really didn't work. It was a technology company working to solve a complex problem, and ultimately technology dominated the discussion. Ultimately, we ended up building something that the business side was not happy with, which made selling it difficult.

An indication of Mistake #2. Neither Jeff nor I had the power, real or perceived, to simply change direction. The Board was supportive of this management structure. This was also a mistake. Mistake #1.

A Real Product versus a Science Project

We talked about "release early/release often," but were scared of looking like idiots in front of major Wall Street and hedge fund clients.

Is it better to wait a bit before releasing to have a more compelling product or to begin getting feedback on a less impressive offering? We chose #1; in retrospect I think we should have chosen #2. By choosing to wait we lost our intimacy with the customer (Mistake #5 again), falling into the classic (as a "green" entrepreneur I didn't know this, but as a seasoned four-year venture investor I know this now) trap of pursuing a "science project," not building a commercially salable product. Dumb. Another problem: technology and product management were effectively bundled together, with the same decision-makers for both.

This was another crucial error, #2 again. Instead of having product management as the advocate for the customer and the product evangelist, we had technology running the show in a vacuum. Huge mistake. This allowed us to perpetuate the science project for much, much longer than we should have. There were no checks-and-balances built into the system. This was a recipe for failure. I intuitively knew it then but as an inexperienced entrepreneur didn't feel empowered to act. Really, really stupid. After 20 years of making consistently good business decisions why didn't I throw a fit and and be more assertive in communicating my concerns? No good answer here.

And these bad behaviors were reinforced by an unplanned event that sharply impacted our psyche: being on the front page of the Financial Times. It is hard to call it a mistake since we didn't seek to get such exposure, but I put it down as Mistake #3.

To be honest, this single fact was a very meaningful factor in our failure. It raised the level of expectations so high that it made us reluctant to release anything that wasn't earth-shattering. It was also catalyst for us raising our last and largest round of capital. So the net effect was that it enabled to raise all this money that kept us far from the customer. Truth be told, we were probably afraid of customers at this point because we didn't want to disappoint them or look bad. Oh, we'd build something they'd love. We just wouldn't show it to them until it was done. Ugh. Just so stupid.

Too Much Money

Too much money is like too much time; work expands to fill the time allotted, and ways to spend money multiply when abundant financial resources are available. By being simply too good at raising money, it enabled us to perpetuate poor organizational structure and suboptimal strategic decisions. Mistake #4. We weren't forced early on to be scrappy and revenue focused.

We wanted to build something that was so good from the get-go that the market would simply eat it up. Problem was, with all that money we hid from the market while we were building, almost ensuring that we would come up with something that the market wouldn't accept. And then there were technology issues that came up along the way, very substantive issues, that because of so much money we simply didn't face into nearly fast enough. And this drove a wedge in the company between those that were more plugged into the market (and felt we weren't building the right thing or addressing the data issues the right way) and those who were building the product (and felt very convinced that what they were building was responsive to the market).

I would almost argue that too much money enabled the other six mistakes to be made again and again and again. Seems counter-intuitive, right? It's not. And believe me, I am super sensitive to this issue now as an investor.

If a company wants to raise significantly more money than I think they need to get to revenue, I push back. Hard.

Investor Expectations versus Market Reality

We raised money based on a vision of a scalable web portal, a tool that would eventually be the web-enabled side of Bloomberg. We never believed we'd replace Bloomberg, Reuters or Thomson for market data and mainstream news, but that we'd eventually become a necessary part of the Institutional investor research mosaic. We were positioned as a technology company, not as an alternative research provider or a services business. And it was the deep belief in Monitor110 as a pure technology company that created a rift between the business side of the company and powerful members of the Board. Mistakes #6 and #7, as you'll soon see.

We did an angel round in the latter part of 2005 followed by an institutional round early in 2006, enough money, we thought, to help us build the new version 1.0 of the product. We then did another institutional round in Q3 2006 to further execute against this vision, because the money was offered to us on a pre-emptive basis and around six months earlier than we were planning to do a raise. The new release would be whizzy, fast, comprehensive and use all that neat technology to analyze unstructured data in real time, and to score each data element by reputation and relevance. Easy to filter, discover and analyze.

Super cool, right? Sure. Problem was, we started out trying to analyze most of the dynamic web (probably up to 100 million sources by now) in real-time, and using technology (NLP, pattern matching, etc.) to do the filtering, indexing and categorization. This was no mean engineering feat. We had a very, very large and complex back-end. And even with this, the quality of the data coming through to the end-user was just not that good. Too much spam, still. Duplicate posts. Sometimes mis-categorized. Difficulty applying our reputation algorithms. Not good.

Those closer to the customer wanted to effectively chuck this approach and to build up a high-value corpus of data from the bottom up, using our deep knowledge of the source universe to assemble a body of data from publishers of high reputation. Really more akin to a "Bloomberg for the Web" than the original product, as the sources would be of high-quality and indexed correctly.

They also wanted to build a research capability, where a desk could generate customized reports for clients leveraging our technology and data. But making this fundamental change to our approach towards data and the business model resulted in a fight. Almost a jihad, where certain parts of the company were vehemently in favor of changing our approach while others said "improvement to the current system is right around the corner."

This could only happen because of Mistakes #1 and #2, where nobody could pound the table and say "this is the way we're going to do it and here's why," nor could the business side simply say "this is what our clients want. This is why we should do it." We were one big, passionate, driven, dysfunctional family.

This argument played out over months and months, and cost us an enormous amount of money. Eventually we did change our approach to data, but it was a fight that spiritually damaged the company and morale and had a financial impact that substantially depleted our coffers.

And in Conclusion

The good news for me personally is that I now invest in a way that actively seeks to avoid the seven deadly sins listed above, and the performance of my portfolio companies bears this out. But I simply wasn't smart enough or experienced enough to see all of these mistakes or to feel empowered to do something about them until it was too late.

I would like to thank all of our investors for having the confidence in us to pursue the Monitor110 vision, and I'm sorry that we weren't financially successful. I'd also like to thank the people with whom I worked during my tenure at Monitor110. Not a bad apple in the lot. Smart, hard-working, highly motivated professionals. They will invariably do extremely well in their post-Monitor110 lives.

The market for alternative information and tools is very, very challenging, and the current market environment isn't making it any easier. But there are clear needs out there that should and will be addressed. I will write a post on the alternative information market at a later time. Thanks for listening.

Original article here: http://informationarbitrage.com/post/698402433/monitor110-a-post-mortem

Startup Failure: NewsTilt

Why we shut NewsTilt down

NewsTilt was a YCombinator-backed startup which aimed to provide services to help journalists become entrepreneurs and earn a living off their work online. It closed down in July 2010, a total of 8 months after it was founded.

A while back, we announced to our journalists that we shut NewsTilt down, only two months since we launched. I think people are interested in why it failed, and there are some interesting lessons in our demise, at least for me.

This piece focuses on my own view of why it failed, as opposed to answering the questions and comments which appeared following the publication of our announcement by Romenesko. I'll probably take some time in the future to respond to critics in the future, and to get back to those who contacted me for comments.

Summary

Following the launch, everything started going to shit, and a huge number of challenges to the success of the company had arisen. The biggest of these were the lack of traction from launch, that we had lost the faith of our journalists, and because there were communication issues between Nathan (my co-founder) and me. This combination also killed our motivation.

As a result, I made a carefully thought out decision to shutdown the company, and return as much money as we had left (about 50%) to the investors. Nathan believed the best thing to do would be to pivot our company, and so I agreed to step down to allow him do that. After some work, he agreed that it was best to shut it down (hence the email above), and we are currently going through the steps of winding down the company, and returning the remaining money to investors.

Problem overview

NewsLabs failed because of internal problems and problems with the NewsTilt product. NewsTilt failed because:

- journalists stopped posting content,
- we never had a large number of readers,
- we were very slow to produce the features we had promised,
- we did not have the money to fix the issues with NewsTilt, and it would have been tough to raise more.

None of these problems should have been unassailable, which leads us to why NewsLabs failed as a company:

- Nathan and I had major communication problems, we weren't intrinsically motivated by news and journalism,
- making a new product required changes we could not make,
- our motivation to make a successful company got destroyed by all of the above.

Overall, the most important of these are that Nathan and I had difficulty communicating in a way which would allow us save the company, and that this really drained out motivation.

NewsTilt problems

NewsTilt wasn't a bad idea, but we certainly faced a ton of problems with it. Most of them could be overcome, but its instructive to go through what they were.

Who are you writing for?

NewsTilt was a news destination website, but we very quickly ran out of news. We relied on journalists posting news, and they stopped posting because they largely no longer believed that NewsTilt was good for them.

Journalists felt that they were writing for us, instead of writing for themselves, for their own brands. How could they feel anything else, since that's the impression we gave them by the design of newstilt.com. The most important part of

NewsTilt–that journalists would have their own brands and own domains–got cut from the minimum viable product in order to make launch date [1]. It was never re-added because of technical issues with Facebook.

As a result, it looked like NewsTilt was trying to be another Huffington Post [2], that is, a news company to compete with existing news organisations. As well as convincing the journalists that they were contributing to us for nothing in return, this had knock-on effects: other venues to which a journalist might sell their news refused to allow the same stories to be posted to NewsTilt. This is obviously the right thing to do if you perceive NewsTilt to be a competitor. If they had perceived each journalist's NewsTilt site to essentially be a personal blog, as we perceived it, we wouldn't have had this problem.

Since we weren't making any money for them, and it appeared to them–correctly–that no-one was even reading their content, there was no earthly reason they would keep writing "for us".

Worse is better

Somewhat surprisingly, the journalists we picked were too good. We made a big deal of only hiring the "best journalists", something we spent a great deal of time getting right. We had a guy with a Pulitzer, one with an Emmy, and overall a great deal of talent writing for us [3].

In hindsight, this may have been a big mistake. The kind of writer we actually needed was one that was hungry to succeed. Someone who would write five pieces a day, and who wanted nothing more than to be a big-time journalist. We needed a young Perez Hilton or Michael Arrington, people who wrote for 18 hours a day in order to make their names.

Instead, we got journalists who were already successful in their day jobs, and who already had families and other commitments. They were checking out the latest thing in news, not hungry to make something of themselves. Why would they be, they already had made something of themselves. Unsurprisingly, they didn't write 18 hours a day, instead just dipping their toes in to try NewsTilt out. They applied, and either never started or posted only a small number of articles.

I think it's important to say that we really failed because of a lack of content. But that was a symptom of having the wrong kind of journalists. All the problems the journalists faced, not writing enough, their distrust of Facebook, their unwillingness to socially promote their work, would not have been problems for a young journalist eager to make a name for himself. If we had the sort of people who gave up everything to succeed at their dreams, these problems could have been blown past. But as established successes in their field, it was unreasonable to expect them to make giant changes for uncertain return.

We never made it clear how hard it was going to be to create an online presence, and so when articles went nowhere, there was little motivation to continue. Building a brand online is akin to doing a startup – it'll take five years. But we failed to prepare them for this [4], and we failed to recruit the people with the kind of that kind of dedication to "making it".

Wrong content

The actual content that the journalists wrote wasn't what we needed either. Content on the site was a very high standard, but it tended to be very long pieces. Long pieces online are difficult to make a success of, as the online attention span is very low.

The problem was that I didn't really know how the journalists should write their pieces, only a vague sense that it was wrong. I also didn't really want to tell them what to write, since they were doing us a favour by writing at all. In fact, we actively told them to write what they wanted. This could have been fine, if they had taken their cue from their readers. But we didn't really know who the readers were, and there weren't that many of them anyway.

It took a while for me to work out that the first thing I did in the morning to sate my internet addiction wasn't going to NewsTilt. I was still going to Reddit and Hacker News. When I did read the pieces, I wasn't terribly interested in them; they were definitely better than what I read in the paper, but they didn't trigger the dopamine receptors that made me want more and more, and I didn't know how to fix it.

Who are the readers?

The fact that we didn't know anything about our readers' demographics underscores another problem: I don't understand

news readers. I certainly wasn't one, and I didn't know many people who really were. My customer development had largely consisted of talking to journalists and figuring out what they wanted. I never really–despite good intentions on lots of occasions–talked to people who loved news about why they loved it. So I was unable to say what was going wrong and why people weren't sticking around.

I could possibly have fixed this at a certain level by giving a greater role to the editors, but I was very uneasy about having editors in the first place. I didn't want to tell the journalists what to write, and I felt that greater input for the editors would have made each journalist's brand less individual [5].

Traffic problems

The major reason the journalists bailed was that we failed them. We didn't deliver the things that we said we would, and we wasted the content they provided.

One part of the service we offered was that we would get the journalists traffic. Whooops! Getting traffic is really really difficult. We completely underestimated how difficult it would be, largely because I'd never had a problem with it in the past. When I've needed to promote some pieces I've written, I simply submitted them to Hacker News and Proggit. However, that doesn't generalise in any way.

In retrospect, it was foolish to offer to do promotion for the journalists. We should instead have built the tools to help them with promotion, and let them do it themselves. We had a couple of ideas of what these would be, but we never built them. We felt that we would understand promotion better if we started by doing it ourselves–standard practise for any company–but it sucked up massive amounts of time, and we never got anywhere with it.

We had no domain experience in promotion, and really had no idea what to do. Worse, we had no idea what to tell the journalists to do. We struck upon the idea that if we had fifty journalists, and they each cross-promoted each other to their social networks, then over time we would get more and more people to read each others' content. Suffice to say that the journalists were not happy, and didn't go along with it.

We had pseudo-hired a social media promoter, someone who had gotten bitten by the startup bug, and was interested in community management, social media promotion, etc. She was pretty good, way better than me, but was still relatively new at it. What we really needed was someone who knew this stuff inside out [6], rather than someone who was learning as they went.

We also missed a few opportunities for some traffic. We fluffed our TechCrunch launch by having the piece posted before the site was even launched. This happened because we were sending out old fashioned press releases in an effort to get old media to cover it, and they needed to go out a day early. Old media didn't cover it, but TechCrunch did, a good 30 hours or so before we actually launched. We got 18,000 hits that day, nearly all of whom saw no news, and probably didn't come back.

Technical promises

We basically over promised what we were going to do for the journalists. We showed them a list of things we were going to build quickly, then built things very slowly. Our technical ability was going to be our differentiator, but our technology showing was pretty poor.

One reason is that we misprioritized things, where the thing that really really needed to get built right now changed daily. Another is that building things take time, and we weren't just building one product, we were building tons of products to be used in different ways. We needed Facebook integration and Twitter integration, to integrate the latest changes from the designer, to automatically admit journalists, and alert the editors upon application, and to actually build the product, to build the social promotion tools, etc, etc, etc. Individually each feature might only take between a few hours and a few days each. However, they build up very quickly, and soon we had features we promised would be ready weeks ago, still sitting in a low-priority slot on our TODO list.

We also suffered from a lack of technical resources. I spent nearly all my time doing CEO things [7], so Nathan was left to do the technical stuff alone. We needed to hire quickly to make up for that, but made a key error with our hiring.

We also made some bad technical decisions, such as failing to choose WordPress, and the whole Facebook thing. The

latter also prevented us from moving journalists to their own domains, instead of being under the NewsTilt banner.

The end of NewsTilt

As a result of the errors above, we managed to alienate our journalists. By the time we ran out of content, it was already too late. The journalists were disillusioned and unhappy, and we did not have a product to prevent that.

I realised this when I found myself reluctant to suggest that people visit the site. Journalists were still excited by the concept, readers wanted to check it out, but I realised that I didn't believe that NewsTilt was a good product for either set of our customers.

There were a few possible solutions:

- fix NewsTilt,
- make a different product, aka pivoting the company,
- close down NewsLabs.
- Fix NewsTilt

NewsTilt was definitely a good idea, just one that we hadn't executed well on. The major flaw was that we couldn't promote the stories well enough, and the journalists' stories went into a black hole where no-one read them. The clear way to fix this would be to hire someone who knew what they were doing in this regard. We had a couple of people in mind who possibly could be brought in, and if it required stepping down as CEO so be it [8].

The second flaw was that Nathan and I hadn't been working well together since about February. There probably isn't any blame to be apportioned, except that we should have tried working together on something before building a company together. So one or the other of us should probably leave the company.

Presumably most other things could be fixed. We could shut down NewsTilt for a few weeks or months while we fixed everything, focusing instead on the personal sites of one or two journalists who were dedicated to making it big [9]. We'd fix our technical mistakes, build the tools we really needed, and relaunch later in the year to take in another 10 journalists. Over the next two years, we'd expand to 1000 journalists, and to 10000 over the next five years. At some point, we'd reopen NewsTilt as a news aggregator, when it actually made sense, instead of when we were too small to do anything valuable with it.

We'd need more money, but the new CEO could handle that. We'd need more tech, but the new money could fix that, and whichever one of us stayed would make it happen. Everything could be fixed.

Well, maybe. This assumes we could find a CEO, and that we hadn't burnt our journalist bridges. It assumes that we could find anyone to give us more money, especially since we launched once and got no traction at all [10]. It assumes that we could do something with the first two journalists, especially considering making them money would be very difficult for the first year at least, probably longer, so we'd probably have to pay them ourselves.

Our motivation was sapped from not working well together, and so our ability to be optimistic was pretty sapped too, especially since one of us would be leaving. Working on NewsTilt had never been the fun that startups are supposed to be [11], and the stress was not being counterbalanced by anything positive. We also weren't all that into news and journalism, so our desire to keep pushing the "mission" was extremely low.

I felt that it was probably possible, though extremely challenging, to make something of NewsTilt or some variation of it. When you combine a lack of interest and motivation with that extreme challenge, I think it's clear that this wasn't a good idea.

Make something else

We each had a few ideas of other products we wanted to make. I quite liked the idea of a Heroku for Python, a competitor to Google App Engine. I also had a few products that I knew I wanted to use, so this seemed like the best time to make them.

However, since we weren't going to work together, whichever one of us stayed needed a new co-founder. To be fair to a

co-founder, you have to be able to move to Silicon Valley, especially since my product ideas were all for early adopters [12]. I couldn't move to California without my wife, and I couldn't get a visa that would allow me bring my wife [13]. Take into account I'd essentially be starting from scratch, and that I'd been working 14 hour days for 8 months and had no motivation left, and you can see why I'm not sure I'd have made a success of this.

Close the company

When deciding how to proceed, we had to consider all the people in the company.

We had one employee, but she was working for free, and it now seemed unlikely that we'd be raising the money to pay her. Our co-founder relationship would have to end either way, so if we decided to part ways we wouldn't have been screwing our partner. The journalists had already stopped posting, and they and we were not convinced that continuing was the best thing for them. The only thing left to consider was the investors.

We felt a great duty to two great sets of investors who had put up their personal money to finance our idea, even though they both kissed their money goodbye when they invested. They had largely invested because we had clicked as people, and we wanted to make sure we did the best thing for them too. We still had $20,000 of the original $50,000 they gave us, and we had the option to return that to them.

They too favoured shutting down the company down and returning what money we had left [14]. They accepted that NewsTilt was not going to work, didn't like some other ideas we ran past them, and everybody at that point agreed that returning what was left was the best outcome for them.

So far, everyone–founders, employees, journalists, investors–were better served by closing down. The final set of people to consider were YCombinator.

YC had consulted and advised us every step of the way. When we had co-founder problems, they gracefully refused to take sides. When we wanted to make a new product, they advised us not to proceed without co-founders, and that we'd need to move to Silicon Valley to be fair to those co-founders. And finally, they didn't expect a cent back, telling us to give all the money back to our later investors. Not once in my whole time at YC did I believe that they valued their investment more than they valued us, and they were OK with us closing down. YC is a class act.

Given the options above, it was pretty clear that closing down NewsLabs was the best option.

Would the NewsTilt model actually work?

Despite everything that went wrong, I'm pretty sure that what we set out to do can be accomplished, though perhaps not by us. It is certainly the product that journalists want, but simply one we were unable to deliver.

The biggest thing it needs is a shit-ton of traffic, and that is not easy to bootstrap. Perhaps that's why bootstrapped technology startups haven't been very successful in media, and why most of the inroads into content startups have come from people with more money, ability to create traffic, or both: Demand Media, True/Slant [15], AOL's Seed.com, or Pierre Omidyar's Honolulu Civil Beat.

It also needs to ability to build great tech very quickly, and lots of it. There were tons of things we were dying to innovate on that media companies are still doing very badly, but we hadn't the money to make them happen quickly enough.

Google could have made this work. I believe that if Google applied the same model they could probably succeed. They have the tech ability, they have the traffic, and they already have a massive news property. They also have have a big problem with whiny news organisation, and an elegant solution would be to kill them off by enfranchising their journalists to be their own bosses.

One of the things we did right at the start was that the journalists trusted us. They may not have trusted us to do everything right or to be successful, but they felt we would do right by them. There are some companies who could probably try to replicate this model and not succeed because they wouldn't be trusted by the journalists; Demand Media for example. AOL is still tinged in the scent of the content mill, but they've hired cleverly and are probably capable of pulling it off. Google has shown it is able to operate transparently and somewhat benevolently, but lots of people don't trust it.

Things we did right

So far, I've tried to present why we failed, which focuses on our mistakes and is by definition pretty bleak. But we did lots of things right too.

For a start, we developed an idea that people wanted. I spoke to lots of journalists and came up with an idea that they were really interested in. Then I talked to more and more of them, developing the idea until it was something that lots of people got behind. To a certain extent, this was easier as an outsider looking in. We didn't really know what the problem was with the industry, so we instead looked at what people wanted, and we really nailed the customer development aspect of this, at least initially.

The thing that the journalists wanted was to be in control of their own destinies. They don't like how their newspapers are essentially fucking up their lives and possibly their pensions, and they don't like the content mill alternatives. They really loved the model that we only made money if they did, and that a 20% cut was the way to go about it.

We were able to convince editors to come on board, and even to lend their names to the enterprise. Doug, Les and Jon helped NewsTilt no end, and I am very grateful to them. But they got behind us because they believed in the mission, and they believed in us, and getting this right was important early on.

Similarly, we got about 150 journalists to sign up to take part. Of those, 27 actually wrote something on the site. That's no small number, and I was delighted in achieving that. We also launched, which is a lot more than many startups.

We did OK on the business side. We got into YC, and we ditched out first idea which was going nowhere fast. Between demo day and me going back to Ireland, 7 days, we managed to raise $50K from two small angels. I had a five minute video interview in the Wall Street Journal, which is going on my resume next to my Google Tech Talk. And when the chips were down, we somewhat bravely [16] decided to shut down, saving time and money for our unpaid employee, ourselves, the journalists, and investors.

Personally, I've learned so much from working on NewsLabs. The largest, for me anyway, was that before I started I still had a bit of that shyness that lots of geeks have; doing the CEO job quickly cures you of that. I don't want anyone to come away from this essay with the idea that they shouldn't do a startup because it might fail. Mine failed, and I still learned more and improved myself more than I probably could have in any other way.

Personal Lessons from NewsTilt

I've made a list of what I've personally learnt from working on NewsLabs. Not every one of these will generalise, but I hope my mistakes are instructive for other founders.

In no particular order:

Lesson: Deeply care about what you're working on

I think it's fair to say we didn't really care about journalism. We started by building a commenting product which came from my desire for the perfect commenting system for my blog [17]. This turned into designing the best damn commenting system ever, which led to figuring out an ideal customer: newspapers. While there, we figured they were never going to buy, and we figured out a product that people were dying to use if it existed.

But we didn't really care about journalism, and weren't even avid news readers. If the first thing we did every day was go to news.bbc.co.uk, we should have been making this product. But even when we had NewsTilt, it wasn't my go-to place to be entertained, that was still Hacker News and Reddit. And how could we build a product that we were only interested in from a business perspective.

This compounded when we didn't really know anything about the industry, or what readers wanted.

Lesson: Don't be too ambitious

NewsTilt would be a great thing to succeed at. If you assume that newspapers were going out of business [18], all the

journalists would become their own bosses, and need something exactly like NewsTilt to help them. As such, there was the potential to be the sole source of news online. Ridiculous, massively ambitious, and very unlikely as a result, but if it worked we'd be billionaires.

Next time, I'm going to make a product which will make me comfortably rich, rather than one with a tiny chance of going supernova.

Lesson: Communicate your idea (and manage it)

Our idea kept changing. The more journalists I talked to, the more I understood what our product should be, and what people needed. Unfortunately, that means I changed [19] our idea all the time.

Worse, I failed to communicate effectively what changed and why. I communicated this badly to Nathan, and badly to the journalists. The latter was difficult to manage–no-one is going to listen to everything I say when it changes regularly–but the former was very important. The result was that Nathan and I never shared a vision of where the product was going, which was one of our biggest problems.

Lesson: Make sure your minimal viable product is viable

We were greatly influenced by the idea of a minimal viable product. Build less, launch, then iterate when you have customers. This is a great idea, but judging when your product is viable is always a tough challenge.

The conventional wisdom is to cut any feature which isn't essential. Ultimately though, if you cut features which make your users feel differently about your product, that's a problem. We cut multiple domains from our MVP, meaning that journalists were publishing under our masthead, which substantially changed how they felt about NewsTilt. They were writing for us, whereas they should have been writing for themselves.

Lesson: Be careful about cool ideas

One of the reasons that switching to domains was cut from our MVP was that it didn't work with our Facebook integration. I was married to this idea that Facebook integration was really important. It was the only way that we would allow people to comment, because it forced them to use their real names. This would mean high quality comments, and great community interaction, and I was convinced that this was essential for our success.

And we could only get real names by making everybody use Facebook to sign in. Absolutely everybody worried about this, but I was convinced. I was totally wrong. It alienated people who didn't like Facebook, including some of our journalists. Worse, it caused people to just not comment, meaning they didn't come back, they didn't engage with the journalists, and they didn't start to frequent the site.

This was at the time of renewed interest in Facebook's privacy policy–they had just changed it and people weren't happy. Every day there were articles in newspapers about how Facebook was doing a terrible thing; there was massive backlash [20]. Our own John Graham-Cumming even wrote a piece called 'the Facebook Cull', and told us the only thing keeping him on Facebook at all was that we required it.

Man was I stupid. When people asked to signup without it, I told them no. When the people who did sign up were worried about things being posted to their walls, I didn't understand the problem. When readers said they wouldn't signup to comment, I thought it was just a small minority.

After a few weeks, I realised I had made a mistake, and put it on our "nice to have" list. There were a million other priorities, and how important was this, really. So that was another massive mistake. I should instead have moved it to the top priority, in particular because it held back domains which should (um, also) have been our top priority.

From a technical perspective, it also prevented us from rolling out one-domain-per-journalist a lot sooner. There is an issue with Facebook where single-sign on doesn't work across domains, so readers would have to approve each domain separately. As a result, we didn't introduce what was probably the most important feature for actually making the journalists feel like they were writing for themselves.

Lesson: If you think you should build it, not buy it, you're wrong

We built our whole platform ourselves. Now, we used lots of scaffolding, built on Rails, hosted at Heroku, using every plugin we could find. I reasoned that the platform was the core of our technology, and we were a technology company, and smart technology companies needed the flexibility that comes from writing the core of their platform themselves. In retrospect, this could only be considered premature optimization.

The natural thing to do was build on WordPress instead, but I wasn't having any of it. The major problem with WordPress is that it's written in PHP [21]. I hated PHP with a passion, and couldn't fathom building my company on it. How would we attract good developers? How could we live with ourselves?

Really, I shouldn't have worried. It was far more important to just get it built, and nothing could have helped that more than just using WordPress. We could easily have given journalists distinctive styles, so they didn't feel like they were writing for us, and we could have built things really quickly by just plugging them together.

If I was worried about how productive we'd be with PHP, well, it's not like we had to build everything in PHP. We could just have done data collection in Python, and made it available to the rest of our app through either a web service, the database, or some other way. We might not have been happy, but we would have stood a much better chance of being successful.

Lesson: Build quickly, little company

The biggest fallout of building our platform ourselves was that we couldn't build quickly enough. When you roll your own infrastructure, everything takes time, more time than you can afford. And we had promised the journalists that we would very quickly build a large list of features, none of which were produced nearly quickly enough. This was the major cause of disillusionment–we overpromised and underdelivered–and this was an important reason why.

Lesson: Hire well

Since we needed to build so quickly, as soon as we got some money we wanted to hire another technical person. Nathan had a friend he wanted to hire, who was exactly the kind of great programmer he could work well with. However, it took some convincing to get him to try working on a news website, and he wasn't sure he'd stick with it. We were sure we'd be able to convince him to stay, and we even waited two weeks for him to move to work with us.

Unfortunately, we were never able to excite him about the project, and we quickly realised he was never going to be intrinsically motivated the way we need for a first employee. There was a point when I was over in Cambridge with Nathan and the other developer, and I noticed that the developer wasn't working on a Sunday. Now, we aren't the kind of people who think our employees owe us 90 hours a week, but startups need that kind of work ethic from very early employees–exactly the reason that intrinsic motivation is so important. If your first employee doesn't love what you do, doesn't wake up each morning dying to work on HIS product, you have likely chosen poorly, and that's exactly what we did.

Similarly, we hired someone who wanted to learn how to do community management and social media promotion, instead of someone who knew how already. This is a pretty tough area, and I think we made a mistake in not hiring someone with much more experienced for such an important role.

Update: The comment about working on a Sunday probably raised more ire than anything else in the piece, largely from developers, so I think I should clarify some things.

Firstly, we're not talking about death marches or prolonged periods of 100 hour weeks. It was early days in his employment, maybe the first or second week, and we had either just launched, or were coming up to launch, and had a great deal of things to be done.

Secondly, early stage startups are not normal jobs, and early stage employees are not normal employees. Your first employee is almost a founder. While they get less reward as a result of taking less risk, the success of the company depends on them a great deal.

Thirdly, it's nothing that Nathan and I didn't do ourselves. We worked 80 hour weeks basically since December. And you can get more done in 80 hours than you can in 40, so long as you don't prolong it.

Finally, I'm not speaking ill of the developer. The problem is that we tried to convince someone to join a startup who wasn't really interested. The fault was ours.

Lesson: Distributed teams are hard

At the end of YCombinator, I moved to back to Dublin, and Nathan moved back to Cambridge. Neither of us had US visas, and we both had things to keep us here: Nathan's girlfriend, and my soon-to-be wife. Both of us prioritised our significant others over the company, and I stand by that.

However, that meant a number of sacrifices, including that we were in different cities. We already had a communication problem when this happened, but this made it far worse. Our social media optimizer was in San Francisco, our designer was in York, and we struggled to make the whole thing work.

While there is nothing wrong with remote teams, I think your company has to be at a certain point for it to work. Everyone has to be on the same page, everyone's roles have to be certain, and the communication has to be constant. We had none of these, and we never had the time to implement them.

Lesson: Work with co-founders before starting a company together

You need a co-founder who gets you, and who you work together well with. When Nathan and I signed up together, we had not spent any time working together, and that was a big mistake. Nathan is certainly a great coder, but when we didn't share a vision, and we found it so difficult to communicate, there was no way we were going to get this built.

When I get another co-founder, I'm going to make sure that we spend a lot more time working together on other things before we start a company together.

Lesson: Transparency is tough

It was important to the journalists that we were a very open and transparent company. From the start, we tried to put as much information out there as we possibly could, and the most efficient way was to put every journalist we accepted onto a mailing list. However, this meant that our blunders and critical feedback were visible for all those journalists to see. Lots of them hadn't started writing, we didn't know them, and they had simply signed up, so we were always aware that our emails were semi-public. As a result, when we decided to close up shop, our closing down email was "leaked" to Poynter, leading to all sorts of speculation.

It takes a lot of time to be open like this, and a lot of effort to communicate effectively. The lesson here isn't so much that we did it wrong, but that it's difficult to do well.

Lesson: Don't do too much at once

I finished and submitted my PhD thesis a week before the YCombinator application deadline. Three days later I gave a talk to 900 people at StackOverflow London. When I moved to California in January to do YCombinator, I had still to organise my wedding in May, and I had a paper to write in between. My PhD defence in April was 4 hours after we launched NewsTilt. In May I got married and went on honeymoon.

Basically, life happens. There is never a good time to start a company, just like there is never a good time to have kids. Certainly entrepreneurship favour those without other commitments, but it seems like nonsense that people with other commitments shouldn't start companies.

While I don't regret doing all those things, I need to stop feeling like I can do everything at once. Everything takes time, and that's time which could be spent on other things which are really important too. In retrospect, I should have delayed either the wedding or YCombinator.

Lesson: Be very careful how you are presented to the press

When I gave my demo day speech to investors, I explained that there were tons of customers out there; in 2008-2009, 30000 journalist had been laid off. When I gave an interview to AllThingsD a few minutes later, Peter Kafka focused heavily on the unemployed part of this. I didn't quite realise the problem–it seemed like a minor detail that he was focusing on a bit heavily–until potential customers kept asking "what about solutions for journalists not laid off". Even though our product was for all journalists, it had effectively been maligned by what I thought was a minor detail.

This also led to people thinking we were going to take advantage of them, and that we were just another content mill like Demand Media. Even when we made it clear that we were only making money if they did–taking a 20% cut–this kept coming up, even with journalists who we had signed up and were using our service.

Update: Peter Kafka spent some time defending himself when he wrote about this post. I feel there is no need for this. It was my pitch that put unemployed journalists in his mind, and I had the opportunity to correct him afterwards. Clearly, the error was mine. Lesson: You have the greatest product on earth, and everyone should be lucky to talk to you

My natural way to network is to chat to someone, develop a rapport, and to set up another chat to talk about the world, current events, and (given time) some business [22]. But even to chat to someone, you need an in: what do I know about that person, what could I say to get chatting. So this is what I did on Demo day.

The way it should be done it in to boldly walk up to them and ask them what they thought of your speech. After a few minutes discussing it, they have to talk to someone else, so do you, that's fine, here's a card, we'll chat later.

Irish people tend to self deprecate. They also don't like successful people, and certainly not people who talk like they're the greatest thing in the world. Silicon Valley is the complete and utter polar opposite. Self deprecation is out. No-one invests in a company who isn't convinced that they are the greatest thing to ever happen. I'm thinking "my company has a great idea, but most companies fail, probably mine too, but we'll certainly try as hard as we can to make it work". Great entrepreneurs never concede that they might fail, and tell everyone how lucky they are to be able to invest in their company.

I was certainly told early on to present ourselves as the greatest thing ever, but I didn't properly internalise it until demo day was over, which was probably too late.

Lesson: Two sets of customers is hard

NewsTilt was a product designed to connect journalists with readers. As such, we had two sets of customers, which means we need to do customer development twice. I spent a great deal of time designing the ultimate solution for journalists, and almost no time on what readers wanted. As such, I didn't really know what to make, or what to say to the journalists about what they should write.

It's tempting to look at the lesson as "don't forget to do both sets of customer development", but I think its many times more difficult to do it twice than to do it once. In the future, I'll certainly be aiming for tools that only need to appeal to one set of customers.

Epilogue:

Update: A final point that should be made is that this is not an attempt to blame anyone. The journalists aren't to blame: we didn't make a sufficiently good product for them. The developer isn't to blame; we tried to hire someone for a startup role who had no interest in startups. No, the only people to blame is us, and more specifically me, since I was at the helm when it all went down. What's next?

While I'm still tying up loose ends with NewsLabs, I've gone and gotten a real job! It's great to take a break from the stress of startup life, and I'm loving working on compilers again.

I've just started a job with Mozilla, and I work on the Javascript engine in Firefox. Its a great job, working with smart people, on product used daily by about 400 million people. Its the sort of job I was looking for when I decided to do a PhD 6 years ago, and the perfect place for a geek to end up.

Finally, I'm looking forward to working on side projects. All projects get put on hold when working on a startup, and most were on hold during the final two years of my PhD. I have to write a scripting language, learn Haskell, read SICP and

Concrete Mathematics, fix the mess that is Autotools, and am currently writing a little language for an itch that needs scratching. I'm also going to start writing a lot more. There are a few NewsTilt observations yet to be made, some lessons from my PhD that were a bit informal to make it into the thesis, and about 15 half-written pieces which I hope haven't bit-rotten since I sketched them.

If you're interested, all things will be posted here, and on my twitter, as soon as I get a chance.

Updates

- I added a note to clarify the situation with the developer we hired. HackerNews went nuts over that!
- I rewrote this paragraph because it sounded like I was speaking badly of the journalists.
- I added a note that Peter Kafka and AllThingsD aren't to blame in any way.
- I added an epilogue where I made it clear the blame lay with me, not anyone named or unnamed in this piece. * I assumed that was implicit when I wrote it, and the piece was more of an effort to discuss the problems we faced, but I believe some people interpreted it as an attempt to deflect blame.
- I said we were returning 40% to investors; it's a lot closer to 50% now.
- Matt Mullenweg asked me about my WordPress problems, and I realized I was really projecting my hatred of PHP onto it. Sorry! I rewrote the WordPress section to reflect this.

Original article here: http://blog.paulbiggar.com/archive/why-we-shut-newstilt-down/

Startup Failure: Diffle

Aftermath

It's now been 5 months since Diffle folded, so I figured I'd write one last post to wrap up this blog. There seems to be a lot of curiosity about what happens to startup founders after their startups die, so perhaps this'll prove informative to people.

Job Hunt

I start work as a UI Software Engineer on Google's Search UI team next month. The resume effect mentioned in Paul Graham's essays seems to be true: my recruiter said that one factor in their acceptance was the experience of having run my own business, even if it did fail. (The others were heavy JavaScript experience and strong knowledge of theoretical CS fundamentals, which apparently is not a combination they get very often.) They thought it would make me a good candidate for leadership roles in the future, which I hope is true, though I'm sure there's plenty to learn on the engineering side before then.

It's not a panacea, though. I also applied to Twitter and didn't even get an interview. Skill fit still matters - being a startup founder is not automatically going to get you a job, though it usually helps. As does all the other common-sense jobhunting advice, like going through a referral, being nice to HR, dressing well for the interview, following up, and knowing your stuff.

It also matters what you did in the startup - at my last employer, I interviewed one candidate who had started his own company but outsourced all development to programmers in Russia. When I asked him what he'd learned from the startup, he said, "That you can't depend on outside programmers. Other people are unreliable." He didn't get the job.

Immediate Aftermath

I didn't realize it at the time, but I was flying when I closed down Diffle - running on pure adrenalin. Part of this was from working with the YC startup, and part was just that entrepreneurship tends to put you into this highly-focused, tunnel-vision state that feels just slightly unreal.

That all crashed down about a month later, which happened to be about a week and a half after I started the job hunt. I ended up getting rejected by FriendFeed, and then told the other companies that I wasn't quite ready to go back into the employee world and needed a few months to figure out what I really wanted to do next.

For anyone faced with winding down a company, I'd highly recommend taking a while off before making any big decisions, and not just the two and a half weeks that I'd initially tried. You're not thinking straight when your startup dies - your perspective may be a bit different in a few months, as might your preferences for what you want to do next.

The corollary to that is to wind up your startup before you're totally out of money, so that you have options for what to do next and don't have to bargain from a place of total weakness.

Anyway, I ended up developing a programming language for those couple of months. I'd actually started this about 3 weeks before Diffle, and then put it on hold for the startup. Whenever people asked me what I'd do if Diffle was a huge success and got bought for enough money to retire on, I said, "Invent a programming language or something." I figured that I still had a good amount of savings and didn't know what I wanted to do, so I might as well get that childhood dream out of the way and see how far I could take it.

As it turned out, I decided that this was better off as a part-time hobby project (remember "If your idea starts with 'we're building a platform to...', find a new idea"? It applies to programming languages as well.) I'm hoping to continue it, as a hobby project, though I don't know how much spare time I'll have at Google.

I also spent a lot of this time stripping off pieces of GameClay, cleaning them up, writing docs, and open-sourcing them. This often took longer than writing the initial code.

I started pursuing other opportunities around October. There was a YC startup that was interested in me as a first employee. A couple friends offered to refer me to big companies (Apple and eBay over the summer, and then Google and

IBM in October). I thought I'd try a cold application to Twitter and Facebook (Twitter ignored me, and I nixed Facebook after realizing I'd probably have to code in PHP). And I had a whole bunch of other startup ideas.

I didn't decide what general direction I wanted to go until very late in the process - as in, last week or so. For a while, I thought about just telling the places I was applying to that I was off the market and doing another startup. Even as recently as this afternoon, literally minutes before formally accepting Google's offer, my mom was telling me about how she'd shared one of my startup ideas with a bunch of prospective customers (teachers - it was education-related) and half thought it was a terrible idea and half absolutely loved it. That's often the mark of a good idea, so I wondered if I was making the right choice.

But if I were to do another startup, I'd be stuck with it for the next 4-10 years, it'd have to be profitable within about 2 to avoid running out of money, and this is all in a very uncertain economic climate. And I have no cofounder, so I'd be doing everything myself until I could afford employees, and then I'd have to build a company culture. There's no fun in that - I might be able to pull it off and get rich, but it'd eat up all of my twenties, probably all my friends, and possibly all my sanity. Not worth it.

Some additional lessons

I keep thinking of other things I wish I'd said in the postmortem but didn't think of. Here're a couple that stick out, though I'm only going to devote a paragraph or so to each.

The big one: startups are out there, not in your head. You get ideas by engaging with the world around you. That can mean potential customers, or experts in the field, or lead users, or friends, or even potential investors. It's really tempting to hole yourself up in your room and code, but unless your idea is spot on (and almost no ideas are), this is a recipe for failure.

Pick a direction and go deep. I tried a couple other ideas after officially folding up, yet never really got into them. If you're like most entrepreneurs, it's easy to get overwhelmed by the number of possible things you could be doing. Ignore most of them, and concentrate only on what you can do better than anyone else.

It's also really important to be frugal. This is more apparent after failure than before: frugality gives you options, and you need them. If you're better positioned than we were, you might even be able to turn failure into victory given enough chances, and cash will give you chances.

Conclusion

I still hope that there's another startup in my future. It looks like it'll now be several years off in my future, though. In the meantime, I'm stoked for Google. I thought the people I met there seemed really cool, as does a lot of the stuff that comes out of there. Maybe I can even give the "intrapreneurship" thing a try.

Original article here: http://diffle-history.blogspot.ca/2008/12/aftermath.html

Startup Failure: Link Management System

6 reasons why my VC funded startup did fail

Sorry for the inconvenience, got slashdotted by reddit. Never thought so many were interested. And no, scaling was not one of the reasons the startup failed ;-) No ssh from my work so it took some time to fix it. Thanks for coming (back).

During the dot com boom I founded a software startup with some friends – with me as the CTO. We developed a software for knowledge management. It was a combination of blogs, wikis, a document management system, link managment, skill managment and more. We started in 1999 which was quite early for wikis and blogs (Moveable Type was announced 2001). The link management system was essentialy the same as Delicious later. Beside all those new ideas (for 1999 at least) there were three great features:

Everything could be tagged. Skills, people, links, documents, blog posts, wikis, something which is today called folksonomies. Tags could refer to other tags to form onthologies. Tags could link to other documents, blog posts, persons. Everything could be rated from 1 to 5.

We had a clever fuzzy search based on tags and ratings. Searching for "people with oracle knowledge" would also reveal experts for SQL Server – for example to stuff your project if an Oracle guru wasn't available.

We've got quite some money as a seed investment from a VC and were quite happy and successfully developing our application. We did show it to many users and received very favorable feedback from big companies.

Why did the startup fail and I'm no millionaire? There are a myriad of reasons, but as I wrote in "Rules for a successful business", the rules for a sucessful business are easy:

- The customer is the most important thing in your business
- The best business plan is to sell people the things they want
- Your business is successful if your earnings are higher than your spendings

So the most important thing is to sell – a fact lots of startups forget. And we did too. After much thought it comes down to these six reasons why we failed (beside the obvious one that the VC market imploded when we needed money and noone was able to get any funding):

- We didn't sell anything
- We didn't sell anything
- We didn't sell anything
- The market window was not yet open
- We focused too much on technology
- We had the wrong business model

In more detail:

We didn't sell anything, Part 1

We didn't sell anything because we didn't have a product to sell. As good engineers we wanted to wait until the product is finished and then start selling. Midway we started selling nevertheless with a nearly finished 1.0. This led to too much focus on development and not enough on sales. Without a finished product we thought we couldn't go to customers and try to sell it. We've slowly learned two things:

You don't need a product to start selling if it's software. The first sales meetings with managment were perfectly possible with screenshots, mockups and slides. As the topic was new to our customers, we first needed to convince them on the concepts (wikis, blogs, tagging) and that could be done without a finished product. Start selling before you found the company. Start selling now! You do not need to have a company to sell new ideas to customers. Start selling now! When people really want to buy your product, start the company.

We didn't sell anything, Part 2

We didn't sell anything because we had no sales person. Bummer. Of course we were looking for one and the business plan said: High priority is finding a sales person for the founding team. This took time and resources and we didn't have that. If you want to sell something, get a sales founder or hire someone from the start.

We didn't sell anything, Part 3

We didn't sell something because customers wouldn't buy. The product was great, the customers favorable, but they took too long to decide. We wanted to sell a knowledge managment tool bottom up to project managers and through them to companies. But everytime a superior heard of knowledge managment he decided it should be on his agenda. So the topic of knowledge managment moved up the chain of command and we had no real decision maker. We talked to irrelevant people (because the topic became strategic fast for our customers) and lost lots of time. Ask people if they are allowed to buy your product. Get to the one who says "Yes" fast. We had several big companies in our sales queue and I'm sure they would have bought in the end, but as a startup we couldn't wait. SAP for example compared to us could have waited and sold the product after 12 months. Selling enterprise software takes a lot of time.

The market window was not yet open

The market window was not yet open. Noone heard of blogs, wikis and tagging. We had to educate our customers on the benefits of wikis (Everyone can edit! Everyone! How dare they!) and blogs (Everyone can have an opinion and write about it! Everyone!) and tagging (They can build ontologies! We need a comittee to define an ontology for everyone! Otherwise chaos will eat us!) Several years later it would have been much easier to sell a blog, wiki and tagging plattform.

We focused too much on technology

All the founders were interested in technology. We've worked with EJBs (not mature back then), we made everything spit out XML and rendered that to HTML with XSLT (not fast enough), wrote our own OR-Mapper – what a stupid idea (Hibernate not available), tried CSS driven websites (not enough knowledge available back then). This lead to rewrites and took lots of our time. Discussions about technology – we should have talked about customers – took time too and led to frustration.

We had the wrong business model

Plain and simple: we had the wrong business model. Selling software can eventually reap lots of money, but it takes time. We had upfront costs, making sales deals took a lot of time and we burned money without income.

The better model would have been: Do consulting on knowledge management and start with an open source product.

We did consulting to companies on how to do knowledge managment, use wikis etc. But we didn't take any money for it, because it was part of our sales process. Focusing on cosulting and billing people would have created a steady income.

I did get into open source later with SnipSnap. SnipSnap took (a small part of) the ideas from the startup (wiki and blog) and was relased as an open source tool. Lots of people downloaded the software and installed it on their desktops. We really made installing SnipSnap easy, so it spread fast.

I've talked to a boss of a very big software and consulting shop and he told me, wikis would never work for them: too chaotic and not structured enough. Well – in fact I knew that there were several SnipSnap installations in his company :-)

As others are practicing now we could have gotten our foot in the door with an open source project, then sell support and enterprise features on top. Companies later paid us money to put features into SnipSnap, make it scale better and for other enterprise thingies. But in 1999 we didn't know as much about software business models as we know now.

What can you learn from my mistakes? Not sure, but start selling. I've learned a lot though about software managment, products, business models, money and being a CTO.

Thanks for listening.

Original article here: https://web.archive.org/web/20131231033505/http://codemonkeyism.com/6-reasons-why-my-vc-funded-startup-did-fail

Startup Failure: PlayCafe

10 lessons from a failed startup

A year and a half ago, my co-founder Dev Nag and I started an internet TV network for games called PlayCafe. Our ambitious plan was to run highly interactive game shows in which everyone was a contestant.

Players could watch our hosts, answer questions, win prizes, form teams, call our studio, live chat, and run their own games. It was a huge undertaking, but despite great engagement — users watched for 87 minutes per session and 40 percent returned within a week — we didn't reach enough users.

We may revive it in the future, but for now, we've placed the site in hibernation and returned remaining funds to our investors.

What follows is a post-mortem of what we did right and wrong and how we will improve next time. I feel too many entrepreneurs are afraid to discuss their failures, locking up important lessons. I hope you find some of this useful.

- Find quick money first. We were fortunate to know several top investors but we spun our wheels pitching too early and trying to optimize terms. Many entrepreneurs seek A-list investors first; quick (but not dumb) money is more valuable. There are only about 30 high-profile investors in the Valley, they all know each other, and they generally have enough deal flow to wait until risk is minimized. Money is more valuable than advice or connections since there are easier sources for the latter.

Next time, we'll focus on strengthening our network of investors who are comfortable at the earliest stage and invest quickly, even if they don't have a high profile (as long as there aren't red flags). More than three meetings is too long and indicates your network isn't broad enough or your pitch is poor.

Lesser-known investors also often have more time to give you. David Shen, previously unknown to us, initiated multiple deals and a site-wide design evaluation for us; it took one meeting for David to commit. You want to find 10 David Shens. Once you have traction, the Reid Hoffmans and Ron Conways will find you anyway.

- Content businesses suck (or: do it for love and expect to lose money). Producing quality content every day is a herculean task, especially live. The idea of creating both the content and technology for PlayCafe seemed achievable, but TV networks focus on distribution and studios on production for good reason: both are hard. Dev and I knew we were production novices but we thought live-filming a pretty girl delivering trivia with one camera guy was simple enough. We were wrong; the business was beyond our pay grade.

Watch American Idol, the country's most popular show, and you'll see how often they screw up despite massive resources: sound and video fail, hosts and contestants stammer, camera angles are wrong, stretches get boring, and it happens despite a reality format that is simpler than live sports or news. They also don't have to deal with DOS attacks, server downtime, scalability, or customer support like we did.

I would advise any entrepreneur or investor considering content to think twice, as Howard Lindzon from Wallstrip warned us. Content is an order of magnitude harder than technology with an order less upside; no YouTube producer will earn within a hundredth of $1.65 billion. This will only become more true as DVRs and media-sharing reduce revenues and pay-for-performance ads eliminate inefficient ad spend, of which there is a lot. The main and perhaps only reason to do content should be the love of creating it.

- Know when to value speed vs. stability. Another reason PlayCafe's complexity hurt us is that developing good content and technology simultaneously required too much time. We tried to make each deep and stable — important, we thought, given our live nature — but we were too slow to iterate in a novelty- and entertainment-based business.

A metaphor I like is that a chess novice can defeat a master if moving twice each round. This generally increases bugs and offends perfectionists, but I agree with Reid Hoffman that if you review your first site version and don't feel embarrassment, you spent too much time on it.

An exception is when your product is mission-critical for users. An eBay outage is a catastrophe while a Twitter one is a

joke. eBay iterates slowly partly because 1.3 million businesses depend on it. It has to get it right.

- Set a dollar value on your time. I agree with Paul Graham that good entrepreneurs are relentlessly resourceful, but I have a bad habit of bargain-hunting for sport. I spent three hours negotiating our wireless bill down $100, which was a poor use of time given our funding. The mantra to pinch pennies ignores the value of time.

Time is arguably more valuable than money because you can't raise more time. Dev suggested pricing our hours. You can divide your available work hours by salary, remaining funding, or total company costs. Ours was around $50/hour. If I was going to spend 5 hours negotiating, I'd have to save at least $250. This value should increase as you gain funding and traction. For anything greater than $500 at any stage, I'd still strive for NPR: Never Pay Retail.

- Marketing requires constant expertise. The main failure of PlayCafe was marketing. Dev and I came from PayPal, a strongly viral product at a company almost hostile to marketing. Our efforts in SEO, SEM, virality, platforms, PR, and partnerships weren't terrible, but drawing users to a live event requires constant, skillful work.

Like creating content, I no longer think marketing is something smart novices can figure out part-time. As the web gets super-saturated, marketing is the difference-maker, and it's too deep a skill to leave to amateurs.

An exception is inherently viral ideas, especially one-to-many virality, where normal use of your product reaches new users, not "word-of-mouth" viral that requires users to advocate you. With inherent virality, a barely adequate product might suffice, though even then marketing should accelerate growth. Next time we'll raise enough to hire a marketing expert early.

- Control and calculate your user acquisition costs. We initially conceived of marketing as a wildly creative exercise: filming viral videos, launching clever campaigns, pitching media players. That's partly true, but the best marketing is controlled and calculated. If you know exactly how much it costs to acquire a user and you control the entire process, you then know how much capital and revenue you need, reducing your marketing plan from fuzzy guesswork to a clean formula.

Until you find a marketing expert, a place to start is the AdWords keyword tool, which shows you how many people Google for certain words, and the Traffic Estimator, which shows the rough cost of buying keyword ads on Google.com. Yahoo has similar tools. The ideal terms have a decent number of monthly searches (10,000+), low number of competing advertisers (3 or less), and strong relevance to your site.

For example, "game TV shows" has 12,100 monthly searches with 7 currently competing ads, while "2008 game show" has 14,800 monthly searches with only 1 ad. The relevance of these searchers is roughly the same so the latter is a better chance to acquire users cheaply. Your first users are the most expensive and can cost $10-20/user, but the cost should decline as your brand and word-of-mouth grows. The promised land is when you convert and monetize well enough to literally buy users.

Other tactics to control and calculate include A/B testing, tracking sign-up and purchase conversion, and creating landing pages to drive SEO and track different campaigns; look at the bottom of Mint.com for a good example. For fuzzier marketing tactics like blogs and press, knowing the time you spend on each, the value of your time, and your break-even acquisition cost will help you calculate efforts that aren't cost-effective. A data-driven culture is a well-oiled machine.

I would also avoid money pits like PR firms, CPM ads, billboards, and TV/radio spots. We wasted $5,000 on a promoter who produced almost no buzz then said it takes a few $5,000 sprees to see results. Unless you control and calculate, these methods are mainly for marketers bad at math.

- Form partner relationships early, even if informal. Two downsides of partnerships are that they're slow and you lack control, but they do have advantages beyond driving users and revenues. Partners can make connections, teach you the market, flag potential competitors, and become potential acquirers. Believing we had little leverage, we de-prioritized several partners who later said they might have bought us if we had built a stronger relationship and proven our value.

We learned that you can build informal relationships with little time. Meeting decision-makers early, keeping them in the loop, and being genuinely helpful builds familiarity and goodwill, which reaps some of the above benefits without the pain of hashing out a deal. It takes foresight and maintenance, but dating before getting married also makes it more likely

the partnership will be healthy.

- Plan costs conservatively and err on the side of raising too much. While I'm doubtful more funding would have made the difference in our case, it would have let us try more tactics. We did a detailed financial plan, but I underestimated costs to fully expand. Next time we'll better know real costs and likely bite the dilution bullet to raise a bit more than needed. This also prevents spending a lot of time raising extra rounds.

Glenn Kelman from Redfin has some nice common costs. To refine, ask a successful entrepreneur for a previous financial plan to vet yours. I disagree with folks who think financial plans are a waste. They are indeed wrong the moment you start, but they help you estimate headcount, fundraising, break-even, and whether your business model is insane.

- The key to negotiating is having options. The single most useful piece of advice I got was from Bill Trenchard, founder of LiveOps: "Always have options."

Almost everything you do as a founder/CEO involves negotiation: closing investors, hiring employees, signing partners, paying vendors, even advocating features internally. The best way to persuade your counter-parties is signaling — implicitly or explicitly — that you have viable options (also called BATNAs). Just two can be enough. Being at the mercy of a lone option is a recipe for getting screwed.

The more humans are involved, the more negotiable the system. If you hear a human say "that's our policy" without much reason, bells should go off that you have room to negotiate if you reach the right decision-maker. Be nice, ask to speak with a manager, and politely signal that your endurance will outpace their patience. Higher-ups know the value of time and make exceptions accordingly. Sales managers are especially persuadable because they're social and work on commission.

That so much of a founder's job involves negotiation also means work can get adversarial and lonely. It really helps to have a group of friends you don't have to haggle with.

- Knowing isn't enough. Most frustrating is that Dev and I knew much of the above going in. We've been doing this for 10 years each across three startups (though this is our first significant one at the helm). I could have sent this to myself two years ago and probably would have thought what many of you are thinking: this is nothing new.

What's new for me is painfully experiencing the gap between knowing and doing. Advice is thrown liberally around the Valley, but like a surgeon who has studied but never practiced, I think it takes a lot of hands-on experience to learn intricacies and exceptions. I think advisers should more often say, "You probably won't get what I'm saying until you screw it up." Expertise takes time, and pithiness comes with a cost.

Plenty of useful advice conflicts for this reason: Know Your Customer vs. Build For Yourself, Don't Raise Too Much vs. Don't Raise Too Little. The better answer to these questions is It Depends. Advice isn't like code that's easily executed, but like map coordinates that require skill and context. My hope is that our experience brings us (and you) a little closer to that.

Original article: http://venturebeat.com/2009/04/29/10-lessons-from-a-failed-startup/

Startup Failure: Untitled Partners

Untitled Partners Post Mortem

For readers who are not familiar with Untitled Partners, the Company was created to enable fractional ownership of fine art. Our hope was to aggregate demand around works of art being sold through the existing channels in the $70B art market, and then to enable our customers to cooperatively purchase and own the art they loved.

On March 9th, we made the difficult decision to shutdown the Company and return almost 50% of the capital we raised to our investors. Over the course of this post mortem, I will try to identify and expose the causes and decisions that led to the demise of Untitled Partners.

1) The writing on the wall:

I can't say that we didn't see an economic downturn coming. When the cover the of the Economist in July of 2007 depicted a well dressed banker standing atop a mountain, with one foot hanging over a steep ledge, the notion that the Dow was going to hit 15,000 seemed a stretch. Fast forward to March, 2008. For a few months, my co-founder and I had been discussing the merits of fractional art ownership as one of a few ideas we were considering. We analyzed the likelihood of success in a slowing economy, and developed a product that we believed could thrive in boom times with an aspirational customer, and in slow times with a conservative consumer. After all, Marquis Jet (a fractional aviation company), had grown from $0 to $1 Billion in revenue on the back of September 11 and the early 2000's recession.

Our analysis was supported by articles in the Wall Street Journal and the NYT, as well as the Mei Moses art index, which suggested the art market was countercyclical and had a low correlation to the S&P. What we didn't account for was the magnitude of the current correction and the effect that it would have on discretionary luxury spending even within a population that financially could still afford our product. We were obviously wrong about Untitled Partners' ability to grow through the subsequent downturn.

What we could have done differently: We met with a lot of industry experts during the vetting of our idea (people with luxury, art, fractional, and consumer expertise).

More questions focused on how our consumers have behaved in previous down cycles would have been helpful. There is a tendency when speaking with sources of information to focus on the largest unknowns and to fill in the knowledge gaps that are preventing forward momentum as efficiently as possible. We could have benefitted from the experiences these guys had in the early 1980's, early 1990's, and early 2000's, those questions just weren't top of mind at the onset of our venture.

2) The timing:

The day we closed the Seed Round with our investors, the Dow closed at 12,063. Five months later, on the day we launched our Company, the Dow closed at 7,552 (having dropped about 4000 points in the 3 weeks leading up to launch). On our launch day, the Wall Street Journal ran a 1400 word article entitled "How to Survive the Art Market Crash". Three and a half months later, the day we decided to wind down our Company, the Dow closed at 6,500. So what were our choices? We had already delayed our launch once, hoping that the New York Times would stop running "End of the World" headlines, but things were worsening, not stabilizing.

The realty was we had built all of the necessary infrastructure to support our offering, we had completed 4 of the 5 milestones we promised our seed investors in our investment deck, and all there was left to do was sell. In a different fundraising environment, I think we might have been able to capitalize the Company on hype vs. revenue, but there was no question that in order to close a new round of funding, we were going to need revenue. So on November 20, we tried to sell. Our offering was met with a barrage of responses to the effect of "I'm not spending money on anything right now, let alone art." The result: we spent a lot more time than I would have expected on "messaging."

How do you say "you just lost half your net worth, want to buy some expensive fine art?" in a way that is appropriate, appealing, and in good faith? Ultimately, we did not find an answer to this question. In that search, however, we stumbled upon one of the decisions I wish we could have taken back (see "Hiring").

What we could have done differently: We could have made a hard push on fundraising ahead of launch, and tried to lock in an additional 12-18 months of runway ahead of performance. I think we would have come up short, as the fundraising environment had already started to deflate by the time we were thinking about this move, but had we capitalized the Company in October, we would have been able to wait things out for a while and see if the picture got any better.

Our angels actually had varying advice on this question. Ultimately, I think it was a better outcome for our shareholders to have failed fast with half of the capital we raised, vs. what the next 12 months would have looked like even with a full tank of gas, but again, I am not an economist.

3) Hiring:

In the face of our inability to find the proper messaging for our offering, we hypothesized that it was due to a lack of marketing DNA within our core team of Andy and I. What we needed was someone to fill this gap, and figure out what we should be communicating to our customer base and how. After interviewing 20 people, we hired a Director of Marketing.

Despite caution from two of our angels, who didn't think it was the right move, we brought on a 3rd core team member three weeks after our launch. She was not capable of solving our problem, due to a combination of a lack of aptitude and a potentially insurmountable task. We laid her off 60 days after she joined our team. This mistake cost us $15k in compensation, another $7K in wasted marketing spend, and an unquantifiable amount of management bandwidth both bringing her up to speed and compensating for her performance.

What we could have done differently: Hiring is hard, and without proper experience, we should have leaned more heavily on our investors to help us with this decision. Hiring was a challenge we found difficult throughout the life of our Company. We made as many bad decisions as we did good ones with regard to hiring full time, part time, and independent contractors/consultants. Biggest takeaway: As soon as the data starts to suggest someone might be the wrong hire, don't wait, immediately start recruiting a replacement, and upgrade as soon as possible.

4) Frugality is a double edged sword

I think we did a great job of being cheap and maximizing the dollars that we spent. In 9 months we spent about half of the $560K we raised. Especially when trying to build a luxury brand, it would have been easy to burn twice as much capital as we did in this timeframe. 90% of the time, we were probably right to be this cheap, and 10% of the time, we may have missed opportunities, or not gotten the minimum quality that we needed because we were so focused on capital preservation. While the VC community (Ron Conway, Sequoia RIP, etc…) was quite adamant about the need to cut burn and preserve capital in these times, I think that advice was probably more relevant to companies (at least in the consumer space) that had established a real footprint in their market. As a very young Company, it is pretty hard to wait out a downturn when you don't have a real market presence to preserve.

What we could have done differently: We developed a pretty solid barometer for spending, but it took a good 3-4 months before spending decisions became second nature. If I look at where the dollars went, an outsized sum was spent on professional services (legal, branding, design, etc…). We could have been more selective about who we engaged in these capacities had we not been running so hard to fill every gap that was holding up our launch. Next time around, greater rigor around reference checks and cost/value analysis will be helpful in managing the costs.

5) Failure to adapt to changing landscape

There are decisions we could have made, directions we could have gone, to generate short term revenue and extend the runway of our Company, but the reality is those decisions would have moved the Company from an interesting venture/breakout story onto a trajectory of small dollar linear growth. I mashed the data around in my head for months, thinking about lowering our price point, moving toward a lead gen for advisory services, building a liquidation platform, and the truth is nothing made more sense than what we were running at (given team, capital on hand, etc.). Would a different CEO with 20 years of experience in our market have seen an angle that I missed and been able to reposition the Company in time? Maybe. I guess I will never know the answer to that question.

What we could have done differently: We could have been more patient and tried to stick around in the art market in any form it would have us. But that is the difference between taking VC money and shooting for the stars, verse maxing out your credit cards and building a lifestyle business. Not much room for singles and doubles in the venture world. We

signed up to disrupt a market and create a category. Every direction I considered going was tempered by an awareness of the opportunity cost of our time and our investors' money. I don't think there are 50 venture ideas in the art market, and especially not in this art market.

6) The Nature of our Mission

While the feedback we got from the business and academic community on the elegance of our model was inspiring, I underestimated the importance of a relationship between our corporate and personal identities. I had heard many times people opine on the importance of a founder being from within the demo he is serving, but didn't really understand why until now. In some ways, we checked that box, but the solution we created was somewhat disconnected from our daily lives or any broader value or mission. Had we been "democratizing the art market," that would have been one thing, but we weren't.

We were not able to live and breathe our product in the way I would have liked to, and ultimately that had a negative impact on our ability to evangelize the Company. We overcame this hurdle, and supplemented our team with DNA that was closer to that of our customers, but we could have done more here with a mission more closely tied to our interests and experiences. I think people like Rob Kalin at Etsy and Vikram Akula at SKS Microfinance have done a great job of building a personal mission into their businesses.

What we could have done differently: Next time around, I'd like to pay more attention to aligning personal and professional missions. As a founder, there is little to no separation between your work and your person, so to the extent your company's mission can fulfill both of those realms, there are synergies to be had. In short, there are lots of ways to make a buck, but few opportunities to achieve something that is truly burning in your mind.

7) So would people have bought art in shares had we launched this business two years ago?

I think so. Doesn't mean we could have scaled it. There were plenty or roadblocks, hurdles, and unexpected irrationalities in the art market that have changed my perspective on the opportunity. I don't think you'll see me rekindle this mission in the next bull market, but I can pretty much guarantee someone will.

8) The people around us

I am extremely fortunate to have gone through this experience with my cofounder, Andy Lawrence. By far the best and most important decision I made through this venture was to recruit him as my partner. His unwavering integrity, amazing discipline, and supernatural persistence were the backbone of our culture. Particularly when you are trying to disrupt a market, you have to run through a lot of walls. Andy was a bull.

We were both fortunate to have a group of extremely supportive and thoughtful investors behind us. I cannot say enough about the importance of having advisors who are willing to pick up the phone and help an entrepreneur think through the challenges and questions that come up everyday. It has been extremely difficult to get over the fact that I lost money for the people who gave us the opportunity to chase down our vision. We are indebted to everyone who took the risk on us.

Original article here: http://www.fabricegrinda.com/entrepreneurship/lessons-from-a-startup-failure/

Startup Failure: Backfence

Co-Founder Potts Shares Lessons Learned from Backfence Bust

We asked Mark Potts, the co-founder of startup company Backfence, to try to set the record straight about why the series of Backfence hyper-local community sites recently closed up shop. What went wrong? What lessons could be learned? In this guest blog post, Potts explains what happened.

There has been a lot of speculation about what went wrong at href="http://backfence.com">Backfence. To date, the company's investors and I have tried to stay out of the second-guessing in the blogosphere and the trade press, largely because there are private business matters involved that we've chosen not to discuss.

Indeed, as with many early-stage companies, some of Backfence's problems were internal and self-inflicted, and actually had little or nothing to do with the many reasons wildly speculated about in industry blogs and in the trade press in recent days.

However, as Backfence's co-founder, I thought it would be helpful to discuss some of what we learned from Backfence — and why I'm still very optimistic that a similar model can and will succeed. As a pioneer in the user-generated, hyper-local field, Backfence hopefully will pave the way for many other efforts to create locally focused online communities that ultimately will become profitable businesses.

Over the past couple of years, we've seen the appearance of a large number of different variations and models for creating and operating user-generated, citizens media or hyper-local sites. Steve Outing wrote an excellent essay in 2005 on the 11 layers of citizen journalism —there surely are more by now. Plus, there is an excellent recent overview of citizens media by Jan Schaffer and the University of Maryland's J-Lab.

WHAT'S ESSENTIAL FOR SUCCESS Like Backfence, all of these nascent efforts are fascinating laboratories — and also like Backfence, none has yet proven to be a successful, sustainable long-term business model. So it's difficult at this juncture to say what's "right" and what's "wrong." But based on the Backfence experience, here are are a few things I believe are essential for the success of a user-generated hyper-local site:

> Engage the community. This may be the single most critical element. It's not about technology, it's not about journalism, it's not about whiz-bang Web 2.0 features. It's about bringing community members together to share what they know about what's going on around town. A top-down, "if you build it, they will come" strategy absolutely does not work, and that's not what Backfence did — otherwise, we'd have launched in a lot more than 13 communities in two years. Instead, Backfence employed a group of talented journalists and community representatives who sought out and interacted constantly with members of each of our communities to encourage them to participate.

Over time, in our more mature communities, this really bore fruit. Could we have done more? Absolutely. It's essential to success. You have to get a critical mass of community participation and eyeballs coming to the site. Also essential: engaging existing local bloggers, Flickr members and other existing local voices. All of these must come together to make a successful hyper-local site. You have to get the community involved. There's no substitute for that.

> It's not journalism — it's a conversation. Actually, it's whatever the community wants it to be. The magic of hyper-local sites, be they Backfence, other startups, Yahoo Groups or local blogs, is that they provide a forum for community members to share and discuss what's going on around town. The back-and-forth of a good online conversation can be as rich, deep and interesting — or more so — than traditional journalism. In fact, the role of journalists in this process is overrated — except maybe by journalists! The less involved site managers are, aside from lightly moderating the conversation, the better.

Proponents of some other citizens media models argue that journalists are essential to hyper-local sites as thought leaders and examples of professional reporting. But that adds considerable expense, and in our conversations with community members we learned that the intrusion of an all-knowing journalist would tend to stifle, not enhance, community conversation, by setting top-down agendas and crowding out community members. Let the audience decide what's important and choose its own leaders.

> Hyperlocal content is really mundane. We heard this criticism all the time. You bet it is — if you're an outsider looking in. To members of the community who actually live with these local issues, it's vitally important. It's precisely that mundane content, and the conversations around it, that brings life to hyper-local sites. I find that when I look at supposedly thriving hyper-local sites, they seem boring and pedestrian to me. Exception: WestportNow.com. Why? Because I grew up there and know the place and its players. It's that simple: It's relevant to me.

From outside, hyper-local content all looks mundane. But it's information and conversation that's important to a specific local audience and flat-out unavailable anywhere else — the far end of the Long Tail. That's not mundane, it's a competitive advantage.

> Trust the audience. We were asked all the time, mostly by nervous journalists, how we avoided having Backfence become a nasty free-for-all. There were many answers: We installed profanity filters, required registration in order to post or comment, asked members to use their real names and we put "report misconduct" buttons on every post and comment. But most of all, we trusted the audience to do the right thing — and invariably they did. All of that is why we can boast that, aside from removing the usual offshore classifieds spam (English bulldogs or capuchin monkeys, anyone?), we very, very rarely had to police Backfence by deleting content. It happened just a handful of times over two-plus years.

The audience took responsibility for what went on at each local Backfence site and debated local issues in a civil manner because it was about their community. And as in their physical community, they were proud of it and took care of it.

> Focus on strong, well-defined communities. We chose the communities in which Backfence launched based on demographics, population density, local governance, commercial viability and competition, among other factors. But as much as any of these, we chose them because they had a strong, well-focused sense of place and community pride: "I live here, I don't live over there."

Moreover, they were well-defined geographically. Beyond a certain size, communities lose their focus. There are too many different governmental bodies, local organizations, schools and people to get a clear grip on what it means to be a community. It's possible to argue, in fact, that a hyper-local site ideally should operate at the neighborhood level, and that even a town is too big. Trying to create a hyper-local site that covers a large area increases the potential population and spreads the focal points of interest too broadly. You only care about the high school your kid goes to; the one across town might as well be 3,000 miles away. It's all about focus: local, local, local.

> Leverage social networking. The rise of MySpace, Flickr, YouTube and the commercial version of Facebook — virtually all of which have happened since Backfence launched more than two years ago — demonstrates the power of social media. Local communities are social beehives anyway. Why not take advantage of existing local connections and the virality and marketing reach of social tools such as member profiles, "friending" tools, widgets and the ability of members to exchange messages with each other? This was an element we unfortunately were unable to get off the drawing board at Backfence, because of business issues and other priorities.

> There is most certainly a robust hyper-local advertising business. Indeed, local advertisers are eager for new online advertising vehicles. I've seen it suggested repeatedly that Backfence failed because it couldn't sell advertising to local merchants. Not true. In fact, we sold ads to more than 400 advertisers, more than any other similarly sized hyper-local effort that I'm aware of. It was clear that we had staked out an affordable and lucrative corner of the local ad market.

Ads in local newspapers (even community weeklies) are too expensive for many small local businesses. Alternatives like the Yellow Pages, Val-Pak-style coupon flyers and local radio are similarly pricey. And most small businesses don't know from AdSense. That presents a ripe target for a talented, hard-working ad sales team concentrating on offering low-cost ads to local businesses who want to reach members of their communities through hyper-local sites. It's a rich, untapped marketplace.

> Keep costs down. To become a successful business, hyper-local citizens media efforts have to be lightweight and lean. Many hyper-local sites today are one-off labors of love, with little or no hope of becoming profitable. Nothing wrong with that, unless you want to make a living at it. To make a successful business, a hyper-local operation has to be run as much more than a hobby. More than likely, the costs (and revenue) need to be spread among many sites to create a successful business model. The Backfence formula averaged about one staffer per community

> site, and in retrospect, that probably was too rich.

Oh, and you don't necessarily have to pay community members to contribute to and participate on the sites. That's not their primary motivation. They want to be seen among their neighbors as sources of local knowledge and opinion, and thatâs more valuable to them than being paid. Indeed, we never had a single member ask for compensation. And when we offered coffee cards in exchange for posts, Backfence users actually rebelled. They thought we were making the site too commercial, even though ads didn't seem to bother them at all — because local ads are local information.

> Partner with a media company or some other distribution source. Because of the critical need to market to and engage the community, it's better to piggyback on a print or broadcast partner's existing community relationships and marketing power. It's very, very difficult to start from scratch in a community and get to critical mass without help. For a variety of reasons that made sense at the time, Backfence chose not to go the media partner route. But as newspapers and broadcasters have become more savvy in the past few months about their need for hyper-local efforts, it makes more sense for hyper-local entrepreneurs to hook up with media partners — or for media companies to start their own, dedicated hyper-local efforts.

> Hyper-local works. You need patience and hard work to embed yourself in a community and become a vital cog in the life of that community. But when a community comes together, it's striking. We saw it happening in Backfence's more mature communities after a couple of years. In fact, even after cutbacks earlier this year left us with no day-to-day community outreach, site traffic and posting activity continued unabated, and even rose. Were they at the levels we wanted? Of course not (they never are). But that proves that once the community gets involved, a successful community site can almost run itself. I said "almost!"

> Hyper-local is really hard. Don't kid yourself. You don't just open the doors and hit critical mass. We knew that from the jump. It takes a lot of work to build a community. Look carefully at most hyper-local sites and see just how much posting is really being done, especially by members of the community as opposed to be the sites' operators. Anybody who's run a hyper-local site will tell you that it takes a couple of years just to get to a point where you've truly got a vibrant online community. It takes even longer to turn that into a viable business. Unfortunately, for a variety of reasons, Backfence was unable to sustain itself long enough to reach that point.

CITIZEN MEDIA CONCEPT IS SOUND

In an entrepreneurial startup, there's a blend between capital raised and a point of self-sustainability. Typically, that takes many years. Even though it raised — and spent — $3 million, Backfence was unable to make that leap. Ultimately, the investors and I determined that the company's fundamental internal problems made it impossible for us to reach that transition point, which is what led to our decision to shut down. It happens; most start-ups fail, unfortunately.

And yes, Backfence was one of those startups. I'd like to thank the talented, hard-working employees who spent so much time trying to make Backfence a success — reaching out to the community, interacting with members, selling ads, keeping the site running and adding new features. They were the backbone of the company, and unfortunately lost their jobs earlier this year because of issues beyond their control. I'd also like to thank our venture-capital investors at SAS Investors and Omidyar Network, who took a big risk on an unproven concept and then took a large financial loss when we were unable to successfully execute on that concept. The investors have been consistently helpful and supportive and even now are continuing to explore ways to possibly revive the company in some form, even if there's no remaining financial benefit to them.

Most importantly, we all believe that the core Backfence concept — user- generated hyper-local citizens media — is sound. Someday soon, somebody will make it work and turn it into a successful business. If there's anything I've learned from Backfence, it's that the power and potential of local communities still is waiting to be tapped. In the end, Backfence was a well-intended experiment that unfortunately was not successful in the form it took and at the time it was launched. I hope many other hyper-local citizens media efforts can learn from what we did and rise in our wake.

We'll see you around the neighborhood.

Original article here: http://www.pbs.org/mediashift/2007/07/co-founder-potts-shares-lessons-learned-from-backfence-bust197

Startup Failure: Q&A Service

A Startup Idea Postmortem: Proof That Good Ideas Aren't Always Good Business

A few months ago I laid a project to rest. It was part of a long list of things that I consider experiments – some of which have done well (like this blog) and many which have not (too many to list). Like any good experiment, I learned some valuable things. This is the postmortem of an idea. I hope you can learn something from it too.

It has been over a year since I first encountered Yahoo Answers. It's a nice service as long as you don't need to ask anything important. I played around with it, but quickly became tired of the trash I had to sift through. There were too many questions like "what do you peeps think of my avatar" and "which chick reading this is the hottest?" Give me a break. Time is the only thing in life I don't have enough of, and I have to waste it wading through this kind of junk? So like any good entrepreneur, I thought about ways to improve it.

During my ruminations on the Q&A process, I stumbled across what I thought was a really good idea – make the Q&A nearly instantaneous. Think about it this way… you probably have friends on IM that you are experts in certain fields, right? I have technical friends that I will shoot of messages to like "do you have a chunk of code that does _?"

It's a quick and easy way to search. But often those friends aren't available. Well, what if I could build a really really large network, and just throw that question out into the void? Could software algorithms determine the best place to route it based on knowledge profiles of the people currently connected to the network? If so, then in theory I could get an answer in just a couple of minutes. And I would pay for that.

Think about it this way. I don't know much about databases, so I could spend two hours debugging a problem, or I could find a database expert who could do knew the answer immediately. Isn't that something worth paying for?

I told the idea to some people, who immediately said that's what forums are for. Well, have you ever tried to get help on open source software in a forum? You have about a 10% chance of getting your question answered.

…we realized we had no early incentive, so we had to come up with something. To do so, we played out possible future scenarios of how the business would evolve, and looked for places to give ownership to users. Users love owning things. We let them own an answer.

We thought about the type of questions that would be asked, and realized that some questions would have direct, simple answers that would be unlikely to change. For example, someone may ask "when is Britney Spears' birthday?" That has a simple answer, and if our service grew popular, it may be asked hundreds, possibly thousands of times a year.

There is no need to connect the asker to a different answerer every time that question is asked – we could just kick it out of a database. So if the asker paid $1.00 for that answer, we don't need to pay the $1.00 to a brand new answerer. Then the idea came – let's pay the person that put the answer in the database.

So if someone asks a question and gets back a good answer, we dump that answer into a database. Every time the question is asked, we kick the answer out of the database and paid the answerer a a percentage of the question price. It's a residual. Answer the question once, get paid indefinitely.

We used this model and returned to the asker side of the equation. We could give our answerers incentives – but what about askers? Why would an asker pay to use our site instead of the competition? For one, we hoped that we could provide higher quality answers for certain types of questions.

Back to my question about top ten Taiwanese OEMs – you can't get that answered on Yahoo, so it might be worth paying for. In addition, we hoped that people would pay more for near real-time answers. Consumers always pay more for convenience. But we needed more. So we decided to extend our residual policy to the askers. If you ask a question, and it gets answered, then ever time it is asked again, the original asker gets 30% of the fee and the answerer gets 70%. The pitch would be that if you asked your question on our site, you may make money on it eventually.

Now we had our model. We had a reason for askers to use our site. We had a reason for answerers to user out site. The next issue to tackle was marketing. How do we make them aware of it? We decided to use blogs. What better way to

expand than to piggyback on an existing network? But back up a second – regular readers will know that I don't like the viral-marketing-by-blogs strategy. It rarely works. Everyone wants to do it, but it isn't easy to get bloggers to write about something. So we turned back to our ideas about web economics – we needed an incentive for bloggers to post the site on their blogs.

We had our programmers build a blog widget. It worked like this. I would go register at the Q&A site and I would specify the kinds of questions that should be directed to my blog. The questions would then scroll in the blog widget I added to my site, along with their reward fees. For example, if you went to the Q&A site and asked "how do I calculate Net Present Value" and you offered $3.25 for the answer, it might appear on Businesspundit.com in the Q&A window. A blog reader could then click on it, be taken to the page, offer an answer, and collect the fee. Then the blogger took a cut.

But here's the best part – we could change the reward fee. Think about it. The answerer does not know what fee was offered, so we could bid it up slowly. For a question worth $3.00, we could first send it to a blog at $1.50. If no one answered in 30 seconds, we could send it to a set of different blogs at $1.75, then more blogs at $2.00, and so on until we hit the $3.00 maximum. That type of price discrimination would boost profit margins.

The next issue to tackle was how to get busy people to use it. That was where we came up with one of my favorite ideas. We built a little client app that could be set to filter questions. So if I'm busy, if I'm working on something important, I could set it to only show questions worth $15 or more, or only show questions about Warren Buffett, or whatever it is that I would be willing to stop work to answer. If I'm bored, or goofing off, show me everything that matches my knowledge profile.

The software moved forward, we got the plan in front of a few angel investors, and we even got some good ideas from the people we pitched. An employee at a major semiconductor company told me there were always looking for ways to connect with the successful entrepreneurs of the future, but they had no way of knowing who they were.

She suggested that her employer may want to sponsor questions. If someone asks about a 8051 microcontroller, they don't pay because it gets filtered to an engineer at this company who would answer it in hopes of beginning to build a relationship. The company would pay us for the right to answer questions. I was ecstatic. But the more we moved down the path, the more I realized the complexities involved with selling answers. Knowledge is a tricky thing to sell, because even experts disagree on some answers. What's worse, most people think they know more than they really do. Look at how many idiots think they know stocks, or programming, or even business.

Nearly everyone thinks they can give good management tips. It is difficult to sell something so… confusing, and we realized it would lead to problems down the road. Yahoo, and most of the other sites, fix this by having people vote on the best answer, but we couldn't post answers in public because that would take away our residual incentives. And anyway, I'm not convinced in the "wisdom of crowds" for anything beyond general knowledge. It doesn't work for domain specific stuff.

We reached the point where we needed to pump in more money, and we did a tough analysis of the business. It was such a fun idea, and everyone who heard it was fascinated. But fascination doesn't always translate into cash, and I've done enough things in life that were fun, cool, and unprofitable.

I began to realize that experts in most fields are usually pretty busy people, and with the exception of technical folks (who spend tremendous amounts of time in front of computers), most other people don't like sitting in front of one, and probably won't answer questions for a few measly bucks.

It's a matter of incentives. If you make 100K a year and have a family, you probably wish for more time over more money – so is $7 really that enticing? Sure, some people would love this idea, but would millions?

I did some more market research, and in the end, with all the competition, the low barriers to entry, and the questionable economics, we scrapped the idea. It's better as an add-on to an existing business than as a stand-alone company. (We still have the code (in Rails), and if you are interested in it, my partners may be willing to let it go to the right person.)

The larger lesson here is that business is ultimately driven by economics, and economics are driven by incentives. Entrepreneurs, and in particular web entrepreneurs, are often driven by a myopia that says "if I like it, so will everyone else."

That's what I thought for a while. Some people say go with your gut. Others say trust the numbers and the research. I say

find an idea where the two line up.

I hope you can learn from the insight into my own thought processes throughout the project. I always seem to learn from reflecting on it.

A key skill entrepreneurs need is the ability to distinguish between a temporary setback and the signs of an idea doomed to failure. I think we made the right decision.

Original article here: https://web.archive.org/web/20140327021809/http://www.businesspundit.com/a-startup-idea-postmortem-proof-that-good-ideas-arent-always-good-business/

Startup Failure: Meetro

Anatomy Of A Failure: Lessons Learned

In the spirit of openness, I write this post on what we did wrong at Meetro – a post mortem of sorts. You don't see this often enough in the startup world even though the majority of startups go belly-up. Hell, there are probably a few today that will go away with a whimper. So much knowledge is lost. If you've had similar experiences, I encourage you to share them over at Lefora.

To those of you not familiar with Meetro, we were one of the first location-based social networks. We figured out where you were physically and then we would tell you else was around you in real-time. You would then be able to instant message with them, check out their profiles, and hopefully meet up. Other functionality included telling you about restaurants close by, media created nearby, and various local information that pertained to your location. We also supported all your various instant messaging protocols (AIM, MSN, Yahoo) and a slew of other social features.

Even with a robust product we simply couldn't capture enough market share. So here are the major problems we had that, in the end, we couldn't overcome. There were, of course, mini fires and random things but every startup goes through those. I have a feeling some of the other location-based startups out there right now are experiencing the same things.

Most importantly, there was a "location problem". It's really hard to grow a product that's 100% focused on where you physically are. Tons of companies have tried this before and most of them have died. We, of course, were cocky and had to give it a try. There was just something so sexy about the idea that you could load up a piece of software and it would tell you about someone nearby who was interesting to you. Someone will crack this and make billions of dollars on it. I can only hope to be involved in some shape or form, since it's an itch that hasn't gone away for me.

A perfect example of this difficulty was our community in Chicago. We launched our product and got all of our friends in Chicago on it. We then had the largest papers in the area do nice detailed write-ups on us. Things were going great. We had hundreds of active users and you could feel the buzz around it. We threw a few parties that continued to support the good mood all around. Hell, our CTO Sam even met his current girlfriend at one of them.

The problem we would soon find out was that having hundreds of active users in Chicago didn't mean that you would have even two active users in Milwaukee, less than a hundred miles away, not to mention any in New York or San Francisco. The software and concept simply didn't scale beyond its physical borders.

I would also cite a theoretical Idaho example. No matter how slick Meetro was, if you opened it up in the middle of Idaho and no one else was nearby using it, the service simply wasn't that interesting. We did do a few things to address this, such as including "random" people and the whole Meetro team in the application. I must say, this did create some great friendships and kept some people around, but in the end it just wasn't enough.

So how do you overcome the location problem? The way I see it, a Meetro-like application will succeed with one of the following strategies.

- A product with a HUGE audience turns it on. I'm talking about a MySpace, Facebook, etc. Something that has a really rabid audience, where it would be a great feature addition. It still wouldn't be easy. You would have to do it properly right out of the gate. There are just tons of privacy and personal issues that come into play when you are talking about physical location.

I remember vividly being pissed that MySpace didn't work with us (read: buy us) after talking with them off and on. I was naive, the more I think about it. It would have unleashed pure anarchy and it wouldn't have been worth it in the short-term. Plus, at the time they were dealing with their own scaling issues and moving over to .Net. However, I still think it's something that should be on the roadmap of one of these big players.

- A company has a really long term vision and builds it out city by city. This is similar to the approach taken by Yelp. They know it's not easy and I know first hand that they are putting a lot of effort into fostering their communities. It's an ongoing job. You simply can't establish a city and then let it fend for itself.

We had this happen to us in Meetro. After we got the Chicago community going, we up and moved our company to Palo Alto. We weren't there anymore to be the face of the community, organize events, etc. While the service continued and had a core bunch of people using it, by no means was it as rabid as it was at its peak. By the time I realized this, it already was a bit too late and we had shifted our focus to getting a community going in San Francisco.

- Someone creates a viral product that grows like crazy but location isn't the core feature. It just happens to be part of the bigger whole and as people use it more and more, they realize that the location piece built into it has become increasingly valuable for them. I don't know what this product is, otherwise I would be building it right now.

Next, the "download problem". This one is obvious now but when we started in 2004, Web 2.0 hadn't quite happened yet and some download apps were still growing (e.g. Skype). Plus we needed a download for the application to work. Having worked with phone carriers for years, I had decided early on that we simply weren't going to wait for the carriers to get their act together on location. They still haven't done it properly 4 years later.

So to get around them we decided we would build out our own location technology. We filed patents and everything. Simply put, we were counting on the continued growth of WiFi. When you launched Meetro we would scan for all the WiFi networks out there. We would then crosscheck what was out there with what we had in a huge global database (it had grown to 4+ million hotspots when we stopped). If it was in the database, then we would do some trilateration to figure out where you were. If not, we would ask you to enter your location. We would save this info and use it later to crosscheck and verify it against similar data.

This data grew quite fast. We averaged around 5 wifi routers found per location, and in dense cities like New York, I remember averaging 9.6. I remember being quite excited since a key part of our business plan was to build an alternate GPS of sorts and use Meetro as a vehicle to gather and refine this data. The best part was that the technology was self-healing, so if some router was replaced or moved our system, we would learn about it very fast.

In the end, though, the dropoff that happened once people had to download and install Meetro was HUGE and didn't help us at all. If I recall, it was something in the 80 to 90% range. It crushed adoption rates.

Lastly, the "realtime problem". This one is similar to the location problem in that if someone wasn't online when you were online, they were no good to you. While the realtime chat aspect of the application made for some really serendipitous meetings, it also made it harder for people to gauge the activity of their communities, especially if they logged in at odd hours, people were set as away, etc.

I can still feel the magic of when I was on layover in the Denver Airport, I met one of our users, and we grabbed a beer. This is what I dreamed Meetro would be about all the time, but those moments were too few and far between. We did fix this in the end but it was too little too late. So, to anyone tackling this problem in the future, make sure you have some type of persistence built-in, be it "people here previously" or "most common visitors to the area" etc.

Compound these 3 problems and the writing was on the wall. So how would I do things differently today?

1. I would wait until location is all clean and dandy with the carriers and build on top of that. Loopt has been doing the huge business development cycle and has been waiting it out for the past few years. It seems as though they're seeing the light at the end of the tunnel. I'm not really following who else is doing this stuff right now, but I'm sure they're out there.

2. I would take our technology, create an API, and plug it into existing applications that have already been downloaded and given low-level access to the hardware our technology needed. I'm thinking about something that would run on top of AIM, Skype or Firefox, for example. Act as a friendly parasite on top of these apps that are already well established. Expect some licensing announcements soon concerning our IP on this front.

3. The other option is to build out something that just allows you to "check in" where you are at. Kind of like Twitter meets location (Dodgeball did this). However, this in my mind compromises the ultimate use case of a Meetro-like concept.

We could have gone about trying to fix Meetro but the team was just ready to move on. Raising money on the flat growth we had was nearly impossible. Plus I knew that in order to keep the tight-knit team we had built together, we needed to shift focus for sanity sake. People (myself included) just felt beat up. We knew that fixing these issues would involve a

complete rearchitecturing of the code, and people just weren't excited about the idea enough anymore to do it right.

Original article here: http://techcrunch.com/2008/05/20/anatomy-of-a-failure-lessons-learned/

Table of Contents