

ADVENTURES OF A FREELANCE SOFTWARE DEVELOPER

by Jeremy Zerr

Blog: <http://www.jeremyzerr.com>

 LinkedIn: <http://www.linkedin.com/in/jrzerr>

 Twitter: <http://www.twitter.com/jrzerr>

HERE'S WHAT WE'LL TALK ABOUT

- ▶ Life as a Freelancer
- ▶ How Did I Get Here?
- ▶ Reasons to Become a Freelancer
- ▶ The 3 Freedoms
- ▶ Freelance Life Cycle
- ▶ Hard Things about Being a Freelancer
- ▶ Q & A

WHAT I DO

- ▶ I build custom software
- ▶ We create web applications, mobile apps, front to back
- ▶ Full time for 5 and a half years, started part time 7 years ago
- ▶ Added another software engineer to the team last year, my first employee

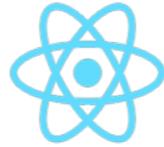


ZERRRTECH

WHO HAVE WE HELPED?



TECHNOLOGIES WE USE



WORDPRESS

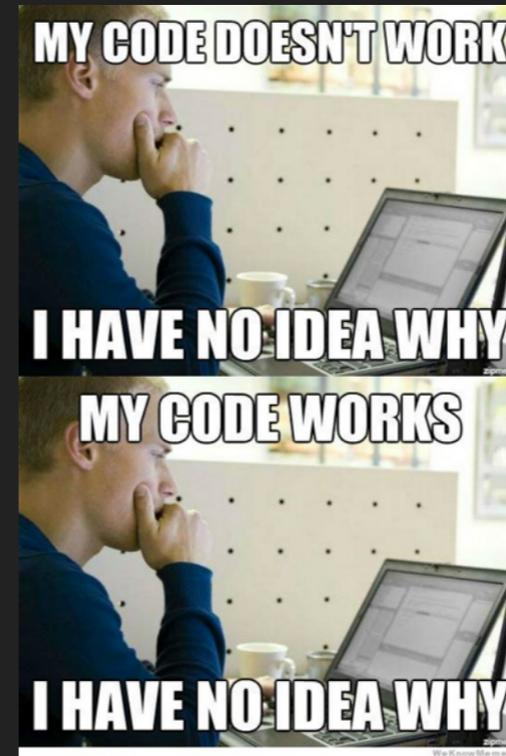


Drupal™



WHAT EXACTLY IS MY BUSINESS

- ▶ Building Software Products
- ▶ Software Project Management
- ▶ Tech Strategy



Some projects involve a little bit of all three. As a freelancer, you get to be much more than just a software programmer. This excites me and is one of my favorite parts about being a freelancer. I like being well rounded and feel this develops me more as a complete entrepreneur than being in more of a narrow, specific role with a software company.

MY PHILOSOPHY

- ▶ Technical Project Management
- ▶ Transparent process with clients
- ▶ Use proven frameworks, leverage open source, be efficient and productive
- ▶ Contribute back to the community - presentations, teaching
- ▶ No outsourcing
- ▶ Never experiment with new tech on client projects

Technical project manager - puts the client directly in contact with someone who is architecting the software and creating it. Ultimate efficiency for the client.

Transparency - A big concern of clients is that you get a project, they pay but won't know what you are working on, or when you are working on it, and will only get to see it when it's done. I share my project management and software development process with the client and involve them. They see where we are at any time, what priorities are, see all code, know when releases happen, get detailed billing statements, and have several direct lines of communication.

Proven frameworks - We use proven frameworks so that if another developer, or an internal developer for our client, needs to work on the code, there is plenty of documentation and well known design patterns to follow to ensure the project is a success even after we are done.

Contribute - We have a choice as software developers to be consumers or producers, we must balance the two. We can just consume open source and never give back, or we can return back as much as we take. We give back in the form of presentations or teaching, help contribute to open source projects and Stack Overflow, and put some code out on Github for all to see.

Outsourcing - We do not outsource programming on client projects to cheap labor, we want to ensure quality control and a team mentality, where we can make the team better and as we always offer the client direct access to technical resources, outsourcing would not allow that. There are way too many competitors anyways, this is how a lot of the big web dev firms run.

Experimentation and learning new frameworks - should be done on your own time, as internal development, and you should develop some projects before you decide to use a framework for a client. This seems obvious, but realize that you are getting money to produce something practical, efficient, and productive, which is very hard to do with a framework that you do not know.

MY EMPLOYED LIFE

- ▶ First job as a fresh Computer Engineer brought me to Boise and Micron
- ▶ Switched positions within Micron, then that department got spun off as a new company called Aptina and moved to San Jose
- ▶ Because of that, off to St. Luke's to move HR/Payroll processes online
- ▶ Ended up as a Clinical Operations Analyst creating a custom web app still being used today



I found myself in a typical situation that a lot of software developers are, working for a big corporation. There was a lot to learn from working for others, if you have the right attitude. Learn about business, investing, what it is like to be an employee, learn what works, learn what doesn't. Focus on making yourself better, every day. And make as many connections as possible. Realize that all of these companies are on my client list, even though I quit all of these jobs, I left on good terms and making sure that people valued my work. Ended up contracting back to them all to help build my business.

MAKING THE BREAK

- ▶ Curiosity - Helping people make money online and building web applications
- ▶ Passion - No matter what my day job was, I always enjoyed and was challenged by creating web apps
- ▶ Risk - What is the worst that can happen?
- ▶ Didn't want to jump in 100% full time if I could go part-time.
- ▶ Don't do it in secret. Maybe somebody can help...
- ▶ How much money did I actually need?
- ▶ First few freelance projects put me from part time to full time

Evaluating risk was a big part. The only real risk I had by quitting or going part time was that if I failed, I would have to go back to a full time job. That isn't a risk at all, that's what most of us coming out of school aspire to anyways. But even as a failure, those experiences never go away, and ends up as an awesome story that nobody else can tell. Even if you fail, it would make you a better, more well rounded employee, leader, project manager, whatever.

As far as money, figure out what your bare minimum costs are. Yeah, it's called a budget. You know it, right? That is your initial goal to get with client work. You are not shooting for 5 days a week of paid client work. You are searching for the number of hours to work to cover your minimum cost. We'll talk about this in the 3 Freedoms section.

TOP REASONS TO BECOME A FREELANCER (FOR ME)

- ▶ Diversity of your skills and who you can help
- ▶ Getting a chance to lead projects, start to finish
- ▶ Develop a broad set of skills
- ▶ Networking
- ▶ Work on newer tech than at product based company
- ▶ The 3 Freedoms



Diversity - doing such narrow, specific work for an employer doesn't help your job prospects. If the only other job like yours is outside of town, you can be put in a tough spot. If you have a widely desired skill set, and you have helped people in lots of different industries, it is both exciting and rewarding to challenge yourself like that.

Lead projects - Experience all aspects of a project, idea, planning, coding, polishing, profit! It's a good pressure to know that you can start and finish.

Broad set of skills - Become a well rounded person and developer. You get to pick projects so you can control your own personal development, that's hard to do in a bigger company. The opportunities are yours to find and accept.

Networking - I've made so many friends and met so many smart people and learned so much over the years. You get to learn about so many companies and businesses, and know what kind of cool stuff is going on in your town. There are all kinds of interesting people out there, not just the people that you work at a company with every day.

Newer tech - New projects mean you get to pick the right tech for the job, and start from scratch instead of having years of legacy code to maintain. Each project is better than the last, and can be a showcase for your best work.

THE 3 FREEDOMS

- ▶ Time
- ▶ Location
- ▶ Money

FREEDOM OF TIME

- ▶ Choose when you work
- ▶ Choose what you spend your time on
- ▶ Choose who you spend your time with



When - Don't like mornings, don't like late nights, no problem. It's always business time!!!

What - Projects are up to you to accept

Who - Who do you want to meet? Who do you want to do work for?

FREEDOM OF LOCATION

- ▶ Choose where you work from
- ▶ Choose your own tools
- ▶ Use whatever gear you want
- ▶ Can use any public services, cloud hosting, APIs
- ▶ Freedom to choose the best solution for the project

Where you work from - home can be OK, but your mind is also programmed to do certain things at home.

Own tools & gear - monitors, desks, color of the walls, no shoes no problem

Public services - not limited to what your corporation will allow inside the firewall

FREEDOM OF MONEY

- ▶ Why do we have to work 40 hrs/week minimum to get our pay?
- ▶ I always wished I could just work less hours for less pay, just enough to cover my needs
- ▶ Goal is to not make more, but to make more per hour, every hour you work on client projects
- ▶ Break Even Days

As far as money, figure out what your bare minimum costs are. That is your goal to get with client work. I was never shooting for 5 days a week of paid client work. I was searching for the number of hours to work to cover your minimum cost, then use the rest of the time to learn what you need, develop your portfolio, and the skills you need.

Then, I would also calculate how much I would make if I stayed as an employee, and figure out how much client work I'd need per week to make that same money. So let's say that ends up being 2 days/week for min, and 4 days/week max. That's your target, somewhere between 2-4 days/week of client work. It's more manageable when you put it that way. Trying to constantly get to 5 days/week of client work will set you up for failure. I don't even do that now. As you can charge more, your time range shifts down.

Also, how much do you value reduced stress and more sleep?

BUILD - HELP - DELIVER LOOP

- ▶ Building Skills
- ▶ Finding Projects to Help on
- ▶ Deliver on the Project

BUILDING SKILLS

- ▶ Leadership/Teamwork
- ▶ Planning
- ▶ Speaking
- ▶ Technical
- ▶ Portfolio

Leadership & Planning skills - non-profits (Habitat for Humanity led team of workers from my church, coordinated volunteers), church (planned a BBQ in a park for 300+ people, twice, organized photo directory and got 200+ families coordinated, scheduled, pictures taken and books published). Have been coaching basketball for last 11 years for 7 & 8th graders. Playing competitive team sports are also great for this. Lead religious education classes for junior high. All non-profits and churches need volunteers, they are not going to be picky, they just need your help.

Speaking - learned a lot from teaching religious ed to 7th & 8th graders. Big challenge to keep them excited and awake. Really learned a lot about preparation, practice, building out reasoning steps, and talking natural & energetic, it's a pretty tough crowd to keep focused and quiet.

Technical & Portfolio - just build. Build, build, build. Don't build client projects for free, if you are going to do stuff for free, pick it yourself and use it to learn new tech and benefit you. Reading books, watching videos, no replacement for hands on building because you also need to build up your portfolio.

FINDING PROJECTS TO HELP ON

- ▶ Goals - How much work do you NEED
- ▶ Dream Client
- ▶ Networking - Have a purpose, help and learn
- ▶ Marketing Yourself - Presentations, Connecting, be a contributor not just a consumer
- ▶ Rates - Test and Negotiate

Back to our discussion on your monetary needs, and your break even days. Keep your client work target reasonable.

Dream client - I've slowly built up a profile of my Dream Client over the last few years, I've gotten more clear about the values I have and those that I want to share with my clients.

Networking - lots of people just don't understand why, think it's just a waste of time. That's because they don't have a purpose. You are there to help and learn. Help people in any way you can, and learn everything about them, their business, what kind of technology they use now and want to use in the future.

Marketing Yourself - I choose to do it via presentations to meetups, connecting with other developers within organizations, and work on relating your past experiences to current problems people are having to see if you learned something that can help them. Always focus on being a contributor, not just a consumer. Always connect with people on LinkedIn.

Rates - The best time to look for new clients is when you have good stable work. Test your rate by going higher than current and see what happens. I prefer to do this in person because you get a better sense about what people are really thinking. Also, be prepared to negotiate, so if you have a target rate, go a little higher, so then if they counter you get something close to your real target and the client also feels like they got something out of the negotiation. When both sides get something out of a negotiation, you can have a great start to the project.

DELIVER ON THE PROJECT

- ▶ Frontend - Estimates & Interviews
- ▶ Middle - Project management software, transparency, time tracking
- ▶ Backend - Invoicing & Maintenance

DELIVER – FRONTEND



- ▶ Harvest for estimates
- ▶ Interview
 - ▶ Prepare, before you even talk with them, you must understand their business, their needs, their tech
 - ▶ I interview them
 - ▶ They interview me
 - ▶ Relate to your past experiences

DELIVER - MIDDLE

- ▶ Software project management
 - ▶ Code - Github
 - ▶ Issues/tasks - Pivotal Tracker, Basecamp
 - ▶ Feedback - Bugherd or just directly into Pivotal Tracker
 - ▶ Harvest - Time tracking
- ▶ Frequent releases, constant feedback, periodic updates
- ▶ Communication and transparency is key



DELIVER – BACKEND

- ▶ Invoicing - Harvest
- ▶ Maintenance - Warranty, hosting, followup

The Hosting business sucks, I used to do it, but almost no cost is worth waking up in the middle of the night to your phone blowing up because a server goes down and getting up to fix is before the client notices. I set most clients up with 3rd party hosting, so that it is still convenient for me to make fixes.

The clients need to own all the assets, the code repo, the hosting, the domain. Don't try and get in the middle and take a cut, it's not in their best interest. Detach yourself so you can work on cool projects. I mean, unless you really like doing web hosting, it's just not for me.

TRADEOFFS

- ▶ Discipline
- ▶ Bubbles of client work
- ▶ Last 10% of a project - Death by 1000 styles
- ▶ Getting first clients and portfolio pieces
- ▶ Taking time to invest in yourself that isn't paid
- ▶ Doing all the other business stuff (accounting, invoicing, taxes, insurance, contracts)
- ▶ Knowing what clients are a bad fit before it's too late
- ▶ Social life
- ▶ Solo coding

Discipline - it's different when you don't have anybody holding you in a desk from 8 to 5, forcing you to work even when you don't feel like it. You have to set goals and expectations for yourself however you need to. I use task lists to say what I plan to accomplish on a weekly basis.

Bubbles of client work - sometimes you get followup work from several clients all at once, I'm not shy about pushing them off. I just clearly tell them I have other client work I've committed to, give them a date I can work with, and make sure you can deliver without stressing out too much. Also, knowing other freelancers/firms, don't be afraid to kick work their way.

First clients & portfolio pieces - Don't do anything for free for clients, but I think it can be wise to discount for those first few projects. Find someone who knows you, build some prototypes to win jobs. Like small prototypes you can build over a fun weekend. That's how I got my first true big freelance project, The Trout Lie.

Clients that are a bad fit - Clients must have money, duh! Warning signs, they must show that they can lead something on their own, and willing to get their hands dirty. If someone calls themselves an idea guy, run!!!

Social life - shared office space, meetups, conferences, lunch/coffee with friends and old co-workers

Solo coding - learn from others by scouring Github for examples in the frameworks you are working on

ANY OTHER QUESTIONS?

CAN I ANSWER ANY QUESTIONS?

- ▶ Clients
- ▶ Marketing Yourself
- ▶ Technologies
- ▶ Rates
- ▶ Hiring an Employee

THANKS!

by Jeremy Zerr

Blog: <http://www.jeremyzerr.com>

 LinkedIn: <http://www.linkedin.com/in/jrzerr>

 Twitter: <http://www.twitter.com/jrzerr>