How To Become a Freelance Software Developer

How to Begin Freelancing

If you think freelancing is for you, there are two basic ways to begin freelancing:

- ***Jump straight away*** into your new adventure;
- ***Make a plan*** to become, or progressively turn into, a freelancer.

The first approach might launch you forward with a burst of momentum, but it can be risky, unless you already have the resources and contacts to start and you know what you're doing.

The second approach is safer, and doesn't usually require a sudden and drastic life change. In some cases, one becomes a freelancer without planning to; it happens as a result of doing more and more part-time work during spare time.

How we got started

Want to see some examples? Here's how some of us got started:

- Matthew Cave had a client approach him unexpectedly with a long-term contract, so he went for it. Slowly over time, it became a partnership, and later a company. But it wasn't planned that way, it started with nothing more than Matthew being tired of his current job, and an opportunity that presented itself.
- Pierre Rochon was less lucky. He had a full time job at the R&D division of a medical company, but he lost it three weeks before his twins were born. He spent a few weeks learning iOS development and then started working on his first project a few months later, for a real estate broker wanting an app to display his listings and allow buyers to contact him directly from within the app.
- Kuba Suder was planning to become a freelancer, but could never seem to get around to it. He worked for a small company in Krakow, doing Ruby/Rails development, but he wanted to become a Mac and iOS developer, and start freelancing as well. Then one day, he was introduced to someone who was looking for developers, for a new project. He decided to quit his previous job and take advantage of the opportunity.
- Ignacio Nieto was in a really well paid job, as solid and stable as one could get in Spain. His salary was great, but he just hated it. Working in a grey office, day after day, with the same faces, in un-creative work nobody cared about, the traffic jams, it was sucking his life. So he decided to take the leap, quit his job and started to work in creative projects, make better use of his time and to take control of his own life. And he never regrets his decision.
- Julio Carrettoni had a day job when he started freelancing. He finally quit his job and decided to become a full time freelancer when he realized that he was making the same amount of money, without all the hassle of being an employee, attending mandatory and useless meetings not related to his projects, and being constantly interrupted with tasks and deliverables not related to his work, etc. He just became aware that he wasn't very productive at his office and the salary wasn't good either.
- Paul Jones instead reached an age where he had to choose whether to stay technical or become a manager, in order to earn more. He didn't enjoy the managerial career path, but he noted that freelancers he worked with did the more interesting work.

- Pawel Krakowiak was looking for a new job when a friend sent him a link to a freelancing platform called RentACoder (now Freelancer). It looked interesting, so he gave it a try, even though he was still looking for a full-time job. After a couple of months, Pawel decided to try freelancing full time. Today he thinks it was the best decision he's ever made.
- Personally, I started that way myself in 2003, doing small projects on RentACoder during my spare time while maintaining a regular job. When I left the job at the end of that year, I started looking for other regular jobs, and in the meantime, I dedicated all of my time to freelancing. I worked on small fixed priced projects, and progressively moved to larger projects, until I realized I could turn that into a full time opportunity.
- Four months later, I made my freelancing career official by opening my sole proprietorship company. And 11 years later, here I am. :] One more thing: being a freelancer made moving from Italy to Poland just a trip by car, with zero hassles regarding finding a job etc.

Finding Projects

Whether you decide to take an immediate plunge or build up slowly, one of the first things you need to do is find some projects to work on. Here's the strategies we've found most effective in finding new projects, in no particular order:

- ***Word of mouth:*** Among our pool, this was the most common source of new projects. As Matthew Cave says, "there's no greater marketing than a happy client." A good reputation is essential here, and I'll talk more about that soon.
- ***Twitter:*** This is similar to word of mouth. Let your followers know you are looking for new opportunities, and gently ask to retweet. Your tweet could be read by hundreds or thousands of people.
- ***Networking at conferences and meetups:*** Another variation of word of mouth, you can meet other developers, potential clients, and more generally expand your network. It can happen quite easily that, by just talking, you get connected to people seeking developers. It's what you would do, isn't it? While discussing, one tells that he's looking for a new project, and you know your old employer needs a freelancer; wouldn't you connect the two of them?
- ***LinkedIn:*** Keep your resume up to date, join groups relevant to your experience and/or preference, write articles, connect with people you know. People will start finding you when they seek developers. Despite I haven't done many of the things I'm suggesting, I still receive offers via LinkedIn, from time to time.
- ***Recruitment Agencies:*** They connect you to employers, in many cases for regular full-time positions, but many specialize in, or include in their offer, contract jobs, which can be performed remotely.
- ***Freelancing platforms,*** such as Upwork, Freelancer and Elance
- ***DAAS (developer as a service) platforms***, such as Toptal, Alt Tab, X-Team and Crew. DAAS platforms are my personal favorite way to find jobs. As I mentioned, I started my freelancing career through RentACoder (now Freelancer), and later oDesk (now Upwork), and today I use many of these platforms, although in the last year, I've mostly worked with Toptal.
- ***Micro-consulting platforms***, such as Airpair, Codementor, Hackhands. Micro-consulting services are a case apart: They consist of providing help in one-on-one sessions, using video calls, screen-sharing and code-sharing. You probably won't make a regular income through micro-consulting alone, but helping others and/or

fixing bugs is one of the best ways to learn new things, and, moreover, you get paid for doing that.

Choosing Projects

If you're using an online freelancing platform, you usually have access to many projects. Don't make the mistake of applying to all of them. It's better to choose a small set of the most compelling—say, five projects—and spend the proper amount of time applying to each of them.

As to how to choose projects, besides the obvious considerations of platform, technologies and languages, there are a few other things to take into account:

- ***Client reputation***: A difficult client can make a big difference in the success of a project and in your ability to do your best work.
- ***Client hire rate***: If a client has posted 100 projects, but never hired a candidate, writing a proposal is probably a waste of time.
- ***Number of applicants***: The more applicants, the smaller the chance you'll be hired. On the other hand, projects with a low number of applicants could require very specific knowledge, or have ridiculously low budgets or other problems.
- ***Feasibility***: It seems obvious, but you have to be sure the project actually makes sense. There's a distinction to make, though: there are cases which are not technically feasible (an app that is capable of

intercepting your thoughts and translating them into text – no technology allows that, yet :]), and cases where it's not technically advisable (an iOS app built using the Android SDK-probably feasible, but not a good choice)

Not all freelancing platforms offer statistics at the client and project levels, to figure out whether a project is a good deal or not. Basing on my own experience, Upwork is the one providing the most comprehensive data, such as:

- the average client rating
- the list of his previous projects
- the number of hires vs number of projects ratio
- the number of applicants to a project, and the average rate or bid for all applicants

I can't say if other platforms provide the same data, but I think they should at least provide alternative ways to let you figure out the quality of a project and the client.

However, when evaluating a project, if your instincts tell you it could be a source of problems, listen to them and keep looking. The same advice applies if you feel you are going to have problems with the client.

Also, be flexible and open to learning new things. Don't bypass a project just because it doesn't use your favorite technology. Think of what's best for the project, not for yourself. Likewise, don't be too quick to dismiss a project because you lack a specific requirement; see "Learning and Experience" below.

Tips for Writing a Cover Letter

When you apply for a project, you'll usually need to write something to explain to the client why she should choose you.

Avoid copying and pasting from one cover letter to another. When you use the same text for every letter, you often wind up talking about yourself rather than about the project. Clients care more about what you can do to solve their specific problems, rather than about your prior experiences.

What you can do is more important than what you've done.

How weird would it be if you brought your car to a mechanic, and she started telling you how good she is at fixing Ferraris, or that she once changed the tires on a Lamborghini? All you want to know is whether she can fix your car, and possibly how much it would cost.

Your skillset and experience are, of course, important, but they probably won't grab the client's attention as much as a direct explanation of what you can contribute to the specific project. Therefore, I recommend you put your credentials at the end of your application.

Be sure that you meet *most* of the requirements set by the customer, at the least. If you don't meet a requirement, it's wise to explain why you would do a good job anyway, or emphasize other contributions you can make to the project that the client may not have considered.

Learning and Your Experience

When applying for a job, you often need to talk to clients about your experience in various areas.

Note that it isn't knowledge of a specific language or framework that determines whether you are an experienced software engineer. Some technologies are more complicated than others and require more time to learn and master. And in your continuing education as a developer, generally you aren't learning *how to do something*, you're learning how to do that something *in a different way*.

As a freelancer in software development, what's arguably more important than experience is the ability to learn quickly and find pointers to the knowledge you need to get something done.

So, when applying for a project, you may omit (avoiding an outright lie) that you don't have experience that fits a specific requirement—if you know that you can quickly fill the gap.

Do you remember when I suggested writing about the project first in your application, rather than about yourself? The point is to *demonstrate* your level of competence, which goes far beyond knowledge of this framework or that language.

If you do that well, the client reading your proposal will realize that you know what you're talking about. Given that positive impression, she may not notice or care that, at the end of your application, you've neglected to mention whether you have the desired five years of Swift experience.

Disclaimer: I'm not encouraging you to deceive your clients. It's about omitting what for you is a minor detail, a discrepancy that, thanks to your experience, you can resolve in a short amount of time without impacting the quality or speed of your work.

In my first medium-large project as a freelancer, I worked on implementing an SMS messaging platform, combined with a simple CMS. The requirements for the project included (among several things) usage of ASP.NET.

The client chose me because I made a thorough analysis of how I was going to implement that system, what tools, which technical challenges I expected, etc.

I omitted mentioning that I didn't have any previous experience with ASP.NET and C#, and he never asked me about that – I just knew I wanted to learn them both, and I was self aware of my ability to complete the project without problems.

The client never realized the omission, until, one day, I told him.

How Much Should I Charge?

Once you find a project, the next thing you'll need to figure out is how much you should charge.

In freelancing networks, *rates* and *reputation* are in fairly strict relationship. If you have no reputation, your clients probably won't accept high rates. If you have a bad reputation, you probably won't be hired at all, regardless of your rates. But as your reputation gets higher and higher, so does the ceiling for your rates. More about reputation in the next article of the series (part 3).

The other important factor in pricing is the country in which you live. I started with cheap rates, about $15/hour, and these days charge an average of $60-70/hour, depending on the project. However, I live in Poland, a country where life doesn't cost a lot. My clients are mostly located in the US, but in the past I've also worked with UK and Italy.

If I were living in the UK, I would probably be earning £60/hour on average, which is about $90. In the US, I could charge as much as $120/hour—or even more, in the right location. I might guess that an iOS developer in San Francisco could ask for two or three times that rate!

Many clients look for the lowest price—but not all of them. There's a good percentage who understand that in life, you get what you pay for. I've frequently written to a prospective client, "I know I'm not the cheapest developer, but…" It's up to you to decide what comes after "but".

Where To Go From Here?

Once you've got a job, you'll want to make sure that you complete it successfully and impress your client – that will make it easier for you to land your next one.

For tips on that, check out our next part of the series, where we'll share some freelance software development tips we've learned over the years.