SIMPLE
PROGRAMMER

The Software Developer’s Guide to Getting a Job

By John Sonmez

Every week, I get emails or comments on my YouTube videos from software developers—some of them quite experienced—who can't seem to find a job.

I have heard every kind of excuse or reason why they can't get one.

Some developers complain that the **older developers have made it too difficult for younger developers to have a chance**—that we've taken up all the good jobs.

Meanwhile, some developers complain that **no one wants to hire older developers** and everyone is ageist.

Others complain that **it has to do with their race, religion, or political affiliation.**

Still others complain that **their skill set is useless**, and no one wants to hire a developer who doesn't have experience with the latest and greatest technology.

While some of these arguments may be true—and, yes, discrimination occurs—**the underlying cause is that these software developers don't know how to get a job.**

At the time of writing this book, **there is a huge demand for software developers**. So many positions go unfilled.

Yet, many software developers complain about not being able to land a job.

How can this be?

How can it be that there are so many vacancies in our industry, yet frequently developers will tell me they have applied for hundreds of jobs and have been rejected by all of them?

The problem lies in not knowing the right way to find a programming job, and I think it's an easy one to solve.

In this report, I'm going to give you **the best tips and tricks** I've learned on how to help software developers get jobs.

The Cookie-Cutter Approach

I'll be upfront with you on this one: I don't like the cookie-cutter approach.

I really don't like the normal way of finding a job because it requires so much wasted effort, and it rarely leads to the best results.

However, rather than just skipping to my preferred way of finding a job, I'm going to go over the standard approach of creating a resume, filling out an online application, and submitting your resume that most software developers—and most professionals in general—do.

The reason I'm going to start here is that **so many software developers are doing this so wrong.**

I've heard countless stories of developers blasting out hundreds of resumes in a single day to every single job posting they could find on Monster.com and wondering why they aren't getting any results.

If you are going to take the standard, cookie-cutter approach, you should at least be doing it right.

I'm going to show you how.

It's a Numbers Game

First of all, you have to understand that the whole process of blindly applying for jobs, the way most people do, is **purely a numbers game.**

It's just like sales.

In fact, **you should really treat the whole thing like a sales process.**

Create a sales pipeline and perhaps even use a CRM system like many sales organizations do to track your prospects through the pipeline.

You'll have something that looks like this:

Jobs applied for > Responses received > Telephone screens > Interviews > Offers

At each stage of the pipe, you should expect only a small number of prospects to make it to the next stage.

For example, suppose you apply for 100 jobs.

Perhaps you'll get 30 responses following up with you in some way.

Out of those 30, perhaps seven will ask you to do a telephone screen.

Out of those seven, perhaps two or three will ask you to come in to do an actual interview.

And perhaps you'll get one job offer out of all that work.

If you want more job offers, you can essentially do two things:

1. **Put more prospects into the pipe** (apply for more jobs).
2. **Increase the percentage of prospects** that make it from one stage of the pipeline to the other.

In other words, you could apply for 1,000 jobs and then get 10 offers, or you could apply for the same 100 jobs and perform better along each step of the way that you also get 10 offers out of it.

To recap, it’s either more prospects in the pipe or better flow through the pipe. That’s it.

If you want the best results, you do both.

How is it, then, that so many developers tell me they’ve applied for hundreds or thousands of jobs and still received zero offers?

Well, there is some luck involved, but most likely it’s one of two things:

- **They are lying,** or rather, grossly overestimating the actual number of jobs they applied for.
- **They really suck** at applying for jobs. Their percentages of prospects moving from each segment of the pipe is really, really low.

Most likely, it’s both.

But have no fear. I’m going to show you how to increase both effectively.

Have a Good Resume

First of all, if you are going to be playing the numbers game, you want to have the highest number of prospects making it to the next level of the pipe as possible.

One of the best ways to do this is to have a really good resume.

If your resume is crap and you get filtered out at the first step, you are going to be wasting a large amount of time, and your overall pipeline is going to be extremely inefficient.

Yes, **a really crappy resume might eventually get you a job**, but the worse your resume is, the more jobs you are going to have to apply for in order for the numbers game to work out.

I don’t know about you, but I’d rather apply for 50 jobs than 5,000 jobs.

One of the first things you should do is to get your resume as polished and effective as possible.

I know some people still disagree with me on this topic, but **I'm 100 percent convinced the best way to do this is by hiring a professional resume writer who specializes in technology industry resumes.**

Yes, **a professional resume writing service might cost you $500** or more, but in the end, that is going to be a small investment for the benefit you are going to get.

One of the best ways you can increase the effectiveness of your resume is to have it professionally written and formatted by an experienced resume writer who knows how to write resumes for software developers.

Create a Quota of Jobs to Apply For

Next, let's tackle numbers.

I like to use quotas to make sure progress is continually made.

Right now I'm writing a new book. I am going to finish the book on time because I have a specific quota of writing at least 1,000 words a day and working on the book 50 minutes each day.

You should have **a quota for the number of jobs you are going to apply for each day** when you are actively looking for employment.

What that number is will depend on what the job market is and how many jobs are available to you, but if you are currently out of work and you need to find a job, it should be at least five jobs a day.

It would not be ridiculous to try and apply for 10 jobs a day.

Figure that each job you apply for will take you around an hour.

What? An hour to click "apply?"

No, an hour because you are not just going to click apply. **You are going to specifically apply for each job by writing a custom cover letter and altering your resume to better match each job.**

Is this a large amount of work?

You bet it is!

Is it effective?

Hell yes!

Remember, not only do we want to increase the number of prospects in the pipe, but we also want to **increase the chances of each prospect moving from one stage to the next.**

So, if you only click “apply” on Monster or some other job site, you’ll be able to apply for a large number of jobs really quickly, but you’ll be sacrificing a large amount of effectiveness.

Also, consider how large the job market is, which is going to be predominantly based on your geographical region.

If there are 50 new jobs per month, you could do a sloppy job of applying for all 50 in one day, but then what would you do for the next week or two?

Sit on your hands and hope you get an interview?

Wouldn’t it be better to apply for five jobs a day but spend an hour or so on each application, greatly increasing the likelihood of success?

How to Apply for Jobs

When you apply for a job, your goal should be to make your resume and cover letter match the job description and any other information you can gather about the company as much as possible.

The idea is that, when the hiring manager or person reviewing the job applications looks at what you submitted, **they are stunned by how perfect of a match you are.**

This doesn’t mean you lie or make things up to put on your resume or in your cover letter.

Instead, you should **try to take your existing experience and highlight the most relevant parts that match the job description**, remove the irrelevant parts (or minimize them), and use some of the words or phrasings that are used in the job description.

Be careful here; you don’t want to just parrot back everything in the job description, because then it will be obvious that you just made everything you submitted match exactly what they wanted.

That being said, **you should match as closely as you can** without seeming too obvious.

Take a careful look at the job description, and get a feel for what the most important skills and characteristics are being looked for in the job.

Modify the ordering and emphasis on your resume to reflect back these points and to show yourself in the best light possible.

Rephrase job descriptions and accomplishments in a way that best shows their transferability to the new job.

Do a little bit of research on the company as well and write up a cover letter that sells the hiring manager on why you are specifically the best applicant for the job. Mention something that only someone who actually did some research on the company would know.

In the cover letter, you should be making a clear connection between the skills being requested for the job and how your experience is relevant and makes you the perfect match.

Yes, this is a large amount of extra work.

It's going to take extra effort.

You are going to have to have several versions of your resume, and each job you apply for will be a further variation of those variations.

But… if you are really serious about getting a job, and you want to stand out, this is what you have to do.

It's easy to click a button and apply for 500 jobs with the same resume and cover letter—that is the lazy thing to do.

Measure Results

One thing you'll want to do as you are applying for jobs in this manner is to measure your results.

You should keep track of **how many jobs you applied for**; **what days on which you applied**; **what response you got**, if any; **what version of your resume** you used; and anything else that might be relevant to your job search.

What you want to do is to have some solid data, so you can see what is and what is not working.

Think of a job search like an advertising campaign.

You have to test various advertisements to see which ones get the best responses.

I’ll often have developers whom I am coaching track the results from the original version of their resume and then track the results they get once they have it professionally written.

Oftentimes, they’ll see a 300 percent or more improvement in the response rate they get from job applications using the new resume.

You need to have this kind of data so that you can alter your plan.

Alter Your Plan

Measuring things is pointless if it doesn’t result in you taking action.

As you are getting back data about how effective your current version of your resume is, the way you are writing your cover letter, or anything else that’s relevant, **you should be using that data to alter your plan and try new approaches.**

Test out a new version of your resume.

Perhaps test a different layout or formatting.

Maybe test using a chronological resume instead of a functional one or vice-versa.

Many software developers trying to get a job who complain to me that they’ve applied for hundreds of jobs knew they weren’t getting any results after applying for 50 jobs, yet **they still used the exact same resume and cover letter for the next 100 jobs they applied for.**

To me, this is straight up lunacy.

How can you do the same thing over and over and expect to get different results?

It makes no sense.

So, **get feedback from what you are doing and alter your plan until you start seeing results.**

Perhaps stick to the same plan for a week and then at the end of the week, go back and look at all the data you have and then decide what you are going to try differently the next week.

If you take this approach, you can almost be sure that you’ll eventually find a job.

The problem is most software developers don’t want to go through all this effort. They just want to complain and say that no one will hire them.

Don’t let that be you.

Work Directly with a Recruiter

Another more traditional tactic you can take to find a job is to work directly with one or more recruiters.

The recruiting industry certainly has its quirks, but as far as applying for jobs goes, **a recruiter can be a big help—**especially if you have marketable skills.

If you can find a good recruiter who has contacts at several different companies, they can help you to fix up your resume to match jobs they are recruiting for, and they can often help you to get an interview with one of their clients.

Here is the thing to remember about most recruiters, though:

Recruiters never want to look bad in front of a client.

When working with a recruiter, you need to convince them you are not going to embarrass them by completely blowing an interview or showing yourself to not be technically competent or a liar.

Do your best to project as much confidence as possible, and a recruiter is much more likely to get you placed with one of their good clients.

Try to think about it from their position. Your performance, good or bad, is a reflection on them and can affect their own professional interests. If you don't conduct yourself competently, a recruiter is unlikely to take a risk on you.

Thinking Outside the Box

Ok, so everything we talked about so far is the traditional, cookie-cutter way to get a job, and it can work if you do it right, but it's not the best way to either get a job or to get a good job paying a high salary.

A majority of jobs—especially the good ones—are not advertised.

Don't believe me? Check out this *Wall Street Journal* article that says as many as 80 percent of jobs are not publically advertised.

How do you get one of those jobs?

You have to think outside the box.

You have to stop thinking that the only way to get a job is to look for a job posting and apply for it.

Yes, like I said, that method can get you a job, but there are perhaps a thousand other ways to get a job.

Don’t be afraid to be creative.

Too many people believe that the standard approach and “best practices” are the only ways to do things, but they aren’t.

There are no rules. There are only guidelines.

It’s up to you to decide how to best accomplish any task you set out to do, including finding a job.

Below, I’m going to give you a few ideas on how to “think outside the box” in terms of job searching, but it is in no way an exhaustive list.

The whole idea behind thinking outside the box is that you do things that aren’t orthodox and aren’t exactly what someone tells you to do.

So, if you are brave and willing to do what it takes, read on.

Build a Network

One of the most effective ways to get access to jobs that aren’t advertised is to know someone who knows someone who’s looking to fill a position. Have a reputation within your network that is so good that as soon as someone hears you are looking, they are jumping all over to try and pull you onto their team.

The only way this is going to happen is if you are willing to invest the time and resources in building a strong network.

There are two main ways that I know of to build a strong network.

1. **Meet lots of people.**
2. **Give them lots of value.**

Most people try and build a network the wrong way.

They wait until they need a job or something else and then they start “meeting people.”

As soon as they meet someone, they start spewing all about themselves and how they are looking for a job.

No one cares what you want.

I’ll say it again.

No one cares what you want.

People only care about what they want.

When you try and network by telling people about what you want, you are doing the opposite of making people like you; you are teaching them to avoid you.

Instead, what you need to do is to meet plenty of people before you ever need anything from them, and when you do meet them, find out the best way you can serve them.

What can you do for them?

How can you help or contribute to them and give them value?

If you do this, you will build a strong network, and you’ll never have to “look” for a job again—I promise.

It takes time, though.

So where do you start?

Well, as a software developer, **I’d be trying to attend as many meetups and software development groups as possible.**

In any geographical area, there are plenty of weekly or monthly groups you can attend.

You will meet developers, recruiters, and managers, and make plenty of other great connections in places like these.

For an added bonus, **offer to present to the group.**

You’ll be able to build up a huge amount of credibility quickly if you offer presentations to the whole group that give them value.

Remember, this is a slow process focused on giving value to others first.

You build a network by investing in it.

You have to sow before you can reap.

If you are patient and attend plenty of events in your community, you will eventually build a powerful network.

Specifically Target Companies

Another major strategy you can employ for finding a job is to specifically target companies you want to work for instead of applying for jobs that companies are posting.

The difference is that you do some research and **you find a company you would like to work for**, or perhaps a few, and you put all your resources and energy into finding a way into that company.

When I was really looking for a remote job as a developer—and there weren't that many—I successfully applied this strategy.

I found one particular company that I knew had a team of completely remote developers.

I learned about the company.

I learned who worked for the company.

I started following the blogs of developers who worked for the company.

I started commenting on their blogs and building relationships with these developers.

The next time they were looking to hire a developer, **guess who multiple developers from that company recommended for the job?**

And guess who got the job?

There are many ways to apply this tactic. This a very general strategy.

In fact, I'll give you some specific applications of it below.

The basic idea is that you pick out the company instead of the job, and you find ways to get into that company.

It might be by **finding people you know who work for the company** or **making new connections** to people working in the company.

It might be by **offering that company something of value.**

It might be by **simply being so persistent** that the hiring managers of that company learn your name and that you won't give up.

A coworker of my wife’s fervently wanted to work for a particular company in Boise, Idaho called Healthwise.

For two years, she applied for just about every single job they ever opened up.

She made friends with employees already working for the company.

She persistently followed up until they relented and eventually hired her.

She’s still working there.

Build Something Useful

One great way to specifically target a company is to build something useful to them and use that to get their attention.

I know several engineers who have been hired directly by companies because they built some kind of tool that used the software the company had built, or they specifically built a tool that they knew the company could use to do their job better.

I’ve heard of designers getting hired because they redesigned a company’s website and sent them the new design for free.

Imagine if you took some popular software you were using, built a new feature or revamped it in some way to make it better, and then presented what you did to the company who built the software.

Now, you’d have to add some real value and you’d have to not just be showing off, but if you created real value for them, for free, don’t you think they’d want to hire you?

Plenty of people find something wrong with a popular website and then tell the company how they would improve it, but **few people actually build a mock-up where they have actually implemented a solution to improve it.**

Start at the Top

Here is a bit of a trick from sales: start at the top.

When most developers reach out to a company to offer their services and try to get a job, they usually end up starting at the bottom.

They might contact HR or a development manager or even an individual software developer, all of whom might have little say over making a hiring decision—especially if there isn't already a requisite for a job.

Instead of starting at the bottom, **try starting at the top.**

See if you can find a way to get in contact with the CEO of the company or even the CTO or Director of Engineering.

One way to do this is to call a published number for the company and simply ask to speak to that person.

Don't say you are a job seeker looking for a job. Instead, say you have a unique opportunity you want to present to him or her, or say you have a business deal.

If you do get to speak to one of these "top dogs," **be confident and have a good pitch.** Don't just ask them for a job. Come up with something of value that you can give them and their company.

Ask them if they could put you in touch with someone who you could follow up with to further discuss what you can do for their company.

Does this sound ludicrous? Insane, perhaps?

You might think this would never work, right?

Well, **that is the whole point of out-of-the-box thinking.** So many people believe they have to start at the bottom or go in the front door.

There is no reason you can't get an audience with a CEO or CTO of a company and have them refer you down to a hiring manager, saying, "I want this guy. Do you have a spot where we could get him in?"

Successful enterprise salespeople use this technique all the time. There is no reason why you can't as well.

Utilize Inbound Marketing

This is the drum I'm perhaps most famous for beating.

I've put out a whole course on How to Market Yourself as a Software Developer that basically teaches you how to utilize inbound marketing to get jobs to come to you instead of you going out to look for them.

I’ve successfully used this technique myself. In fact, **the whole reason you are reading this report is because of the success I’ve had with this technique in my career.**

Countless opportunities have come my way. I’ve received hundreds of job offers, spoken at events all over the world, and **drastically altered the course of my life** because of inbound marketing.

So, what exactly is inbound marketing?

Simple. It’s just **creating content or some other kind of value that gets people to come to you instead of you going to them.**

As a software developer, you can build a blog, make video tutorials, write books, write articles, appear on podcasts, and many other things to build a reputation, get your name out there, and get people—and opportunities—coming to you.

Some people might call this the “get famous” approach, but you don’t have to be famous to make this work for you.

All you need is to be known for one particular thing in the field of software development.

The key is to pick some very specialized area of software development and become the expert in that area.

The smaller the area is, the easier it is for you to become number one, which should be your goal.

Once you’ve established this area of specialty, jobs will come to you. I guarantee it.

Interview the Interviewer

This is another approach that gets you in via the backdoor and gets you right through the gatekeepers.

One great way to get yourself an opportunity where there might not be an open job at the time is to interview the interviewer.

With this technique, you are going to speak to a contact at a company that you want to get a job at, perhaps a development manager, a CTO, Director of Technology, or some other person who would have some kind of hiring decision ability.

You are going to tell this person you are researching information for either an article you are writing or you’d like to interview them for a podcast you are creating.

You could even say that you are recently getting into the field and you'd like to ask an experienced person about the profession.

The idea is to figure out some way you can get into the office and get a conversation with this person.

Most people will gladly jump at the opportunity for free press or the chance to help an aspiring software developer learn more about the field.

Whether you actually end up publishing the interview or not isn't all that important—although I see no reason why you wouldn't since someone else is giving you free, valuable content you can publish.

At this point, you are just planting a seed.

Now you have someone in a company that you'd like to work for who knows who you are and probably likes you since people tend to like people who are interested in their favorite topic: themselves.

In a few weeks, months, or when a job happens to open up at that company, you can follow up.

You can also follow up if you publish their interview on your blog or podcast.

In fact, if you are going to apply this technique, why not interview the CTOs of all the top tech companies in your area?

Then, you can make a roundup post of all the interviews, publish it on your blog, and benefit from some inbound marketing as well.

Be Persistent

Too many people give up way too fast because they don't understand or believe in the power of persistence.

Yes, it's possible to burn bridges by being too persistent and annoying.

However, you have to go way further than you think to get to that point, and even then, you don't really have that much to lose if you weren't going to get the job anyway.

Don't get me wrong. I'm not saying to be overly pushy and that you can bully your way into a job.

What you should be doing is constantly following up, changing your tactics, and pushing forward as hard as you can if you really want to get the best opportunities in life.

Doors won't always open for you. Sometimes you have to take a crowbar and pry the doors open.

Don't give up until you've tried all of these techniques—more than once.

Here's What to Do Next

With the information in this report, you're better equipped to find the perfect job than 95% of software developers.

I've just scratched the surface though of the steps you can take to further your career as a developer.

In fact, I've put together an entire book just for developers who want to grow as professionals and earn more income while doing work they love.

It's called *The Complete Software Developer's Career Guide*.

And I'd like to send you the book—one actionable chapter at a time—for free.

To get started, just go here and enter your information:

https://simpleprogrammer.com/careerguide

About Simple Programmer

Hi, I'm John Sonmez, the Founder of Simple Programmer and author of Soft Skills: The Software Developer's Life Manual.

I created Simple Programmer with the goal of making the complex simple.

My goal for Simple Programmer is not just to help you get a better job and make more money—although I certainly want to do that—but to help you improve in all areas of your life.

To help you build self-confidence, to be more productive, to find and reach your goals, and even to explore the wonderful world of entrepreneurship.

In short, I want to help you succeed and become a better version of yourself each and every day.

SP SIMPLE PROGRAMMER