

Finding Jobs in Software

Harry R. Schwartz

July 19, 2018

I was back at my parents' home in Lancaster, PA last week, chatting with a friend from grad school who's remained in academia, and naturally we got to talking about what advice he could give his computer science students to better prepare them for their probable future careers.

In some later follow-up emails we got to talking about how engineers find jobs. I've fielded this question about a dozen times over the last couple years, so I thought it was about time to crystallize it into a blog post for future linking.

Here are some strategies for finding jobs, ordered roughly from most to least useful:

Friend-of-a-friend networking

Many of the best jobs never make it to the open market at all, and it's all about who you know. This makes sense for employers, since good engineers are hard to find and a reliable reference can be invaluable.

In the case of my current gig at Iterable, for example, a mutual colleague from thoughtbot (a previous employer) suggested that I should talk to Iterable's VP of engineering, since he'd worked with both of us and thought we'd get along well. We did, and I liked the team, so I went through the interview process and took the job.

Like many companies, thoughtbot has an alumni Slack group with a `#job-board` channel. Those sorts of semi-formal corporate alumni networks can definitely be useful, but you'll probably find yourself relying more on individual connections.

“Networking” isn't a dirty word, and it's not about handing out business cards at a hotel bar. It's about getting to know people in a friendly and sincere way, being interested in them, and helping them out (by, say, writing a lengthy blog post about how their students might find jobs). I'm not the type to throw around words like *karma*, but if I were, I would.

Go to (and speak at!) meetups, offer help and advice when you can, and keep in touch with friends and ex-colleagues. In a couple of years you'll have a healthy network. Easy-peasy.

This strategy doesn't usually work at the beginning of a career, of course, but new grads and students should know that it's eventually how things happen.

Applying directly to specific companies

I keep a text file of companies where I might want to work. As I come across companies that catch my eye, I add 'em to the list. When I'm on the hunt for a new job I just consult my list.

Lots of things might convince me to add a company to the list. They might have an especially appealing mission or product, use some particular technology, or employ some specific people that I'd like to work with and learn from.

One shockingly good heuristic that identifies a great workplace is whether a company sponsors or organizes meetups, and *specifically* if they sponsor groups related to minorities in tech. Plenty of great companies don't do that, and they still may be terrific, but if they do it's an extremely good sign.

Job boards

I generally don't use job boards, myself, because I find networking and targeted applications to be more valuable.

The big sites like Indeed and Dice are rarely useful. While some genuinely great companies do cross-post jobs there, there are so many atrocious jobs mixed in that I don't bother with them.

However, smaller and more targeted job boards can be really handy. Someone has created a job site for any given technology (language, framework, database, whatever). If you're really interested in working with a specific tool or in a particular market niche, it might be worthwhile for you to track down the appropriate board.

Similarly, if you're interested in remote work, there are a few boards that cater specifically to that. We Work Remotely is a prominent and reputable one.

The enormously popular tech news site Hacker News posts a monthly "Who's Hiring?" thread (an example). HN focuses mainly on startups and is almost adorably obsessed with trends, tech-wise, so it's a thoroughly biased sample, but it's still a huge selection of relatively high-quality jobs. Browsing it can also give you an idea of what technologies are currently in vogue. Some folks have also built sites that make it easier to filter those listings.

Recruiters

These are the folks that message you on LinkedIn. Recruiters fall into two categories: internal and external.

An *internal* recruiter is an employee of a specific company and hires engineers to work for that company. They're almost invariably non-technical, but they

often have a fairly clear idea of what technical skills they're looking for. They have no idea who you are, or what your goals are, but they're encouraged to find a good fit for the company and are generally harmless.

It's normal to work with an internal recruiter as part of the application process at a software company, especially a larger one.

An *external* recruiter works independently or for an agency. They're market makers; they have a stable of companies who have contracted with them to find employees, and they get a placement fee for every person that one of those companies hires. As such, they have incentives to make as many matches as possible as quickly as possible, and they rarely have to deal with the fallout if the match isn't a good one.

In my experience they add nothing to the job search process and, at best, just gum up the works as unnecessary middlemen. Less reputable ones may edit your resume without your approval, forward it along to companies that you'd never want to work with, and otherwise mangle your reputation. I avoid them.

Helpful and ethical external recruiters are a bit like UFOs. I'm prepared to acknowledge that they might, possibly, exist, but I've never seen one myself or spoken directly with anyone who's encountered one, and I've only heard about them through confusing and doubtful chains of testimonials (and such testimonials usually make me question the testifier more than my assumptions).

University career services

I've never found these to be of any use. The software job market is extraordinarily specialized, and it's virtually impossible for a career services employee (who needs to be able to place every sort of student in every sort of job) to be familiar with it.

A recruiter, whose purview is limited to the software world, will often try to estimate good matches by looking at resume keywords like "Python" or "natural language processing." A university career services employee needs to rely on even more amorphous keywords like "software" or "programming." It's hard for a non-technical person to distinguish a job engineering compilers from one hooking up printers.

Exceptions exist, of course (MIT and Stanford, for example, have predictably excellent software-specific career services), but they're *thoroughly* exceptional.

There are plenty of other ways to find jobs, of course (job fairs at good industrial conferences—like PyCon or Strange Loop—aren't bad, for example, though I've never taken a job through one). But the avenues above are the most common ways that job-finding happens. Good luck!