White Paper

# How to Find and Hire the Best Software Engineers

INTRO

As with many modern technical fields, software engineers are in high demand. Companies around the world need them, so the best talent is snapped up so fast it's astonishing. Of course, you don't want your company to hire the dregs, right? It's better to go about the recruiting process the right way, to attract and hire the best candidates and keep them around.

Table of Contents

![ContractRecruiter.com logo]

Understanding Developer Demographics

The first thing you need to know before you start is that, while "developer" is a specialized field, it's still quite broad. Developers, especially full stack developers, can assume many different roles such as web development, mobile development, software engineering, database or systems administrators, and more.
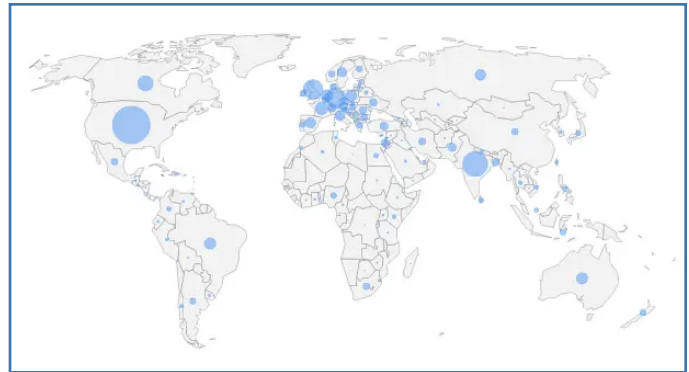

Stack Overflow is a large site where developers can join a network, help one another, show off their projects, and generally participate in a broader community. They perform an annual survey of their audience, with over 90,000 respondents to **their 2019 survey.** This can tell us a lot about the demographics of developers, though it's not entirely representative of the field as a whole.

- The majority of developers responding to the survey were located in the United States, Europe, and India, with Australia, Canada, and Brazil also representing sizable numbers of developers. Nearly every country in the world had at least a few respondents, however, proving that developers can originate anywhere.

- The most popular specialties amongst developers are web development, with full-stack developers making up the most common category. Desktop and enterprise developers come in second, with mobile developers third. Other popular roles include database and systems administrators, designers, and data scientists.

- Over 40% of survey respondents have under five years of professional experience coding. This doesn't mean they're not talented, just that many of them are young or have only recently brought a hobby into their professional lives.

In terms of pure demographics, amongst professional developers, over 70% are white or of European descent, with South Asian, Hispanic, and East Asian following. This is more of a sign of the overall underrepresentation of PoC developers more than anything and can present an excellent opportunity if you look for candidates on a global scope.

Check out the survey; while it's not entirely representative of developers as a whole, it's a good idea of the kinds of people you're likely to see most frequently in your search for qualified software engineers.

Building a List and Sourcing Qualified Candidates
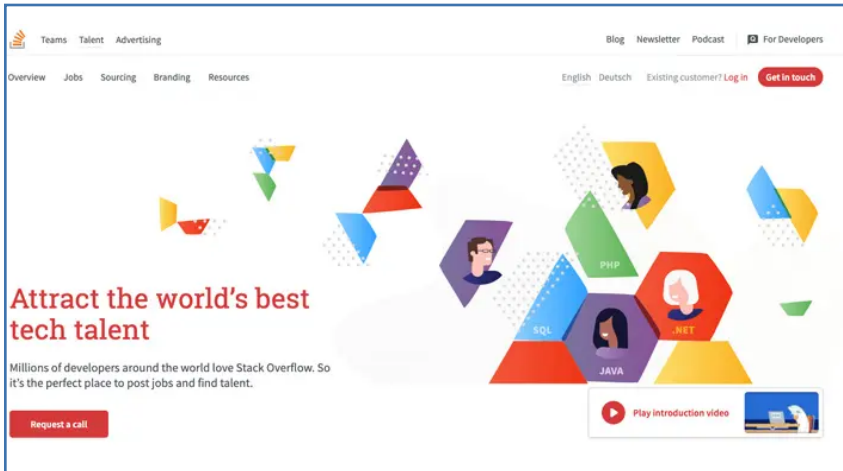
Every good recruitment process begins with building a candidate pool. Your eventual hires are only as good as the people in your pool, so you will always want to find the best, most qualified candidates to populate that pool. The better your candidate pool, the faster you'll be able to go from interview to hire, and the more satisfied both you and your new hires will be with their career.

Where can you look to source candidates to fill your pool? Here are the best sources beyond "a job portal" or your website's career page.
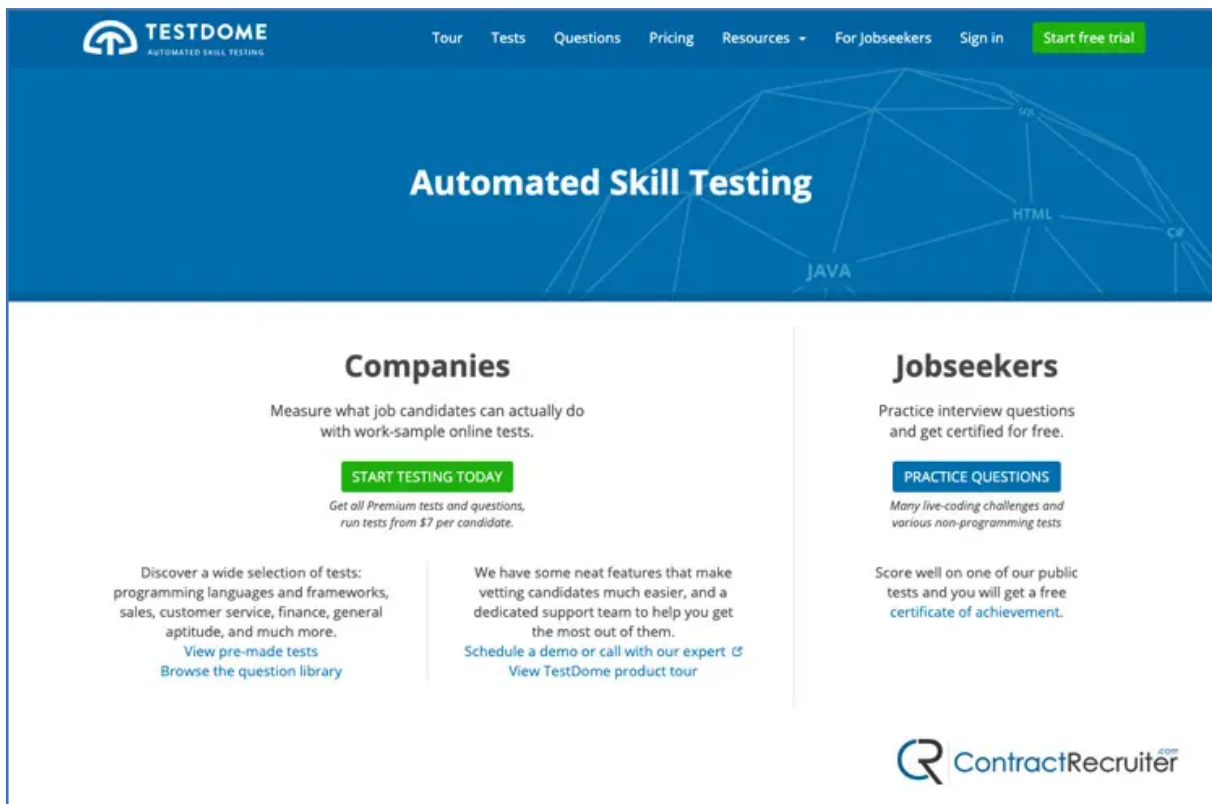
- Personal referrals. High-quality candidates tend to associate with other high-quality workers, and can recommend people they know would be a good fit for a role. Your existing employees likely have a better grasp of what they need in a coworker than your hiring department, as well.

- Technical blogs. Many professional and talented hobbyist software engineers maintain blogs where they discuss and show off their projects. Finding candidates who demonstrate talents publicly gives you an opportunity to reach out and invite them to apply.

- Open Source portals. Open-source software is increasingly popular, and the people who contribute to it have publicly visible proof of their talents. Whether you're looking at individual projects on GitHub or professional contributors to large open-source companies, there are plenty of candidates out there.

Of course, you're not limited to these channels. Traditional channels can bring in plenty of qualified candidates as well. The above are simply great ways to recruit passive candidates and pull talent from many sources.
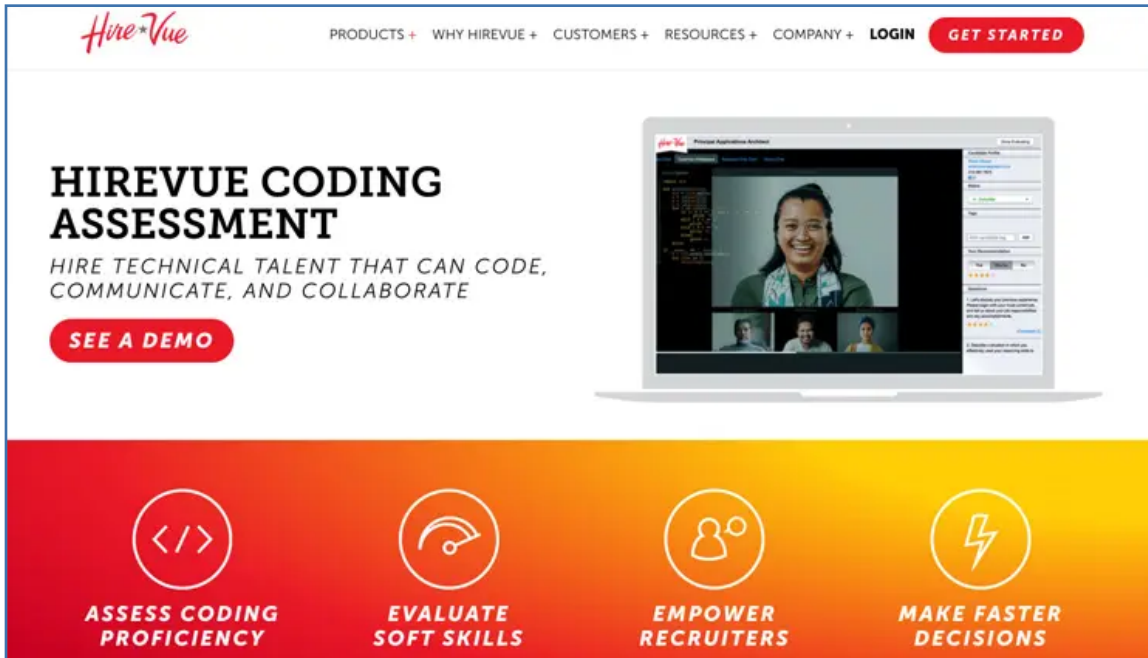
Making Use of Code Assessments

A good software engineer needs to know how to code. A great software engineer needs to know how to code as part of a team. You can test both as part of your interview process by using a combination of traditional interviews and code assessments.

Code assessments are tests you can develop or purchase from companies that develop them. They can range from simple to complex and are meant to test a candidate's ability to code on specific projects. The idea here would be to work with your current lead developer, or whoever heads your development team, to identify tests that assess relevant skills. For example, if you need a software engineer skilled with web development, you want to avoid simple language-based testing.

TestDome provides a wide variety of tests on different subjects, ranging from PHP, Java, and C to mobile tech, data science, and software engineering. Questions range from free and simple to complex and paid, with pricing based on the number of candidates you want to test.
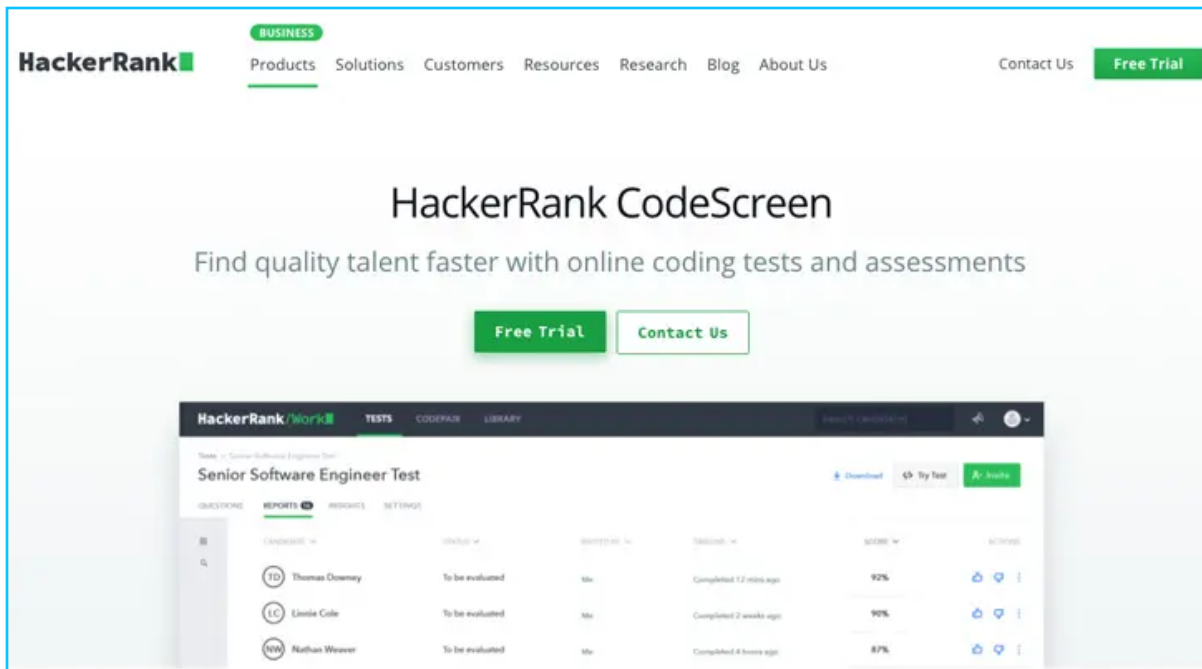
HireVue also offers code assessments, including coding language proficiency and soft skill analysis tests to evaluate candidates from all angles.

HackerRank offers, among other things, CodeScreen. CodeScreen is a code assessment platform you can use to deliver tests to candidates for them to work on at their leisure, giving you data not just on their performance but on how quickly they complete the project.

Of course, you can always develop tests of your own and deploy them in a manner of your choosing. All it takes is a member of your team to develop the test and to judge the results.

Determining a Software Engineer's Technical Abilities

```
18   /**
19    * @Class
20    * The base address class that can be used to signify any kind of address.
21    */
22   class Address {
23       /**
24        *
25        * @param {string} address1 The primary address line.
26        * @param {string} address2 The secondary address line.
27        * @param {string} city The city where the address resides.
28        * @param {string} state The state where the address resides.
29        * @param {string} zipCode The zip code where the address resides.
30        */
31       constructor(address1, address2, city, state, zipCode){
32           this.address1 = address1;
33           this.address2 = address2;
34           this.city = city;
35           this.state = state;
36           this.zipCode = zipCode;
37       }
38
39       (method) Address.getReadableAddressString(): string
40
41       Get's a readable address string.
42
43         • EX: 1234 Test LN. STE 310, Indianapolis, IN 46237
44
45       @return — The address string.
44       getReadableAddressString(){
45           return `${this.address1} ${this.address2}, ${this.city}, ${this.state} ${this.zipCode}`;
46       }
47   }
```

Speaking of judging the results, how can you do that? The obvious answer is to talk to an expert, whether they're your current lead engineer or someone you bring in from the outside.

That said, there are a few ways a non-technical person might judge technical abilities, albeit with a grain of salt.

- Review multiple answers to the same question from different applicants. Do any of them stand out as more elegant than others? Watch for issues like "reinventing the wheel" by writing code to do something a function in the language can do inherently.

- Issue a programming challenge with no language requirements. The choice of language and the accuracy of the resulting answer can say a lot about the candidate developing it.

- Try to break the code. If you ask a software engineer candidate to design a simple applet that performs a function, what happens if you were to add strange inputs, like letters instead of numbers, special characters, or code snippets? A good engineer should (as long as it's part of the assignment) program to avoid these kinds of issues.

Also, while it might seem obvious, consider issuing at least one skills test either in person or via a live meeting. On the job, a software engineer should use all of their available resources to develop quality software, but you want to make sure they aren't wholly relying on external sources for their answers.
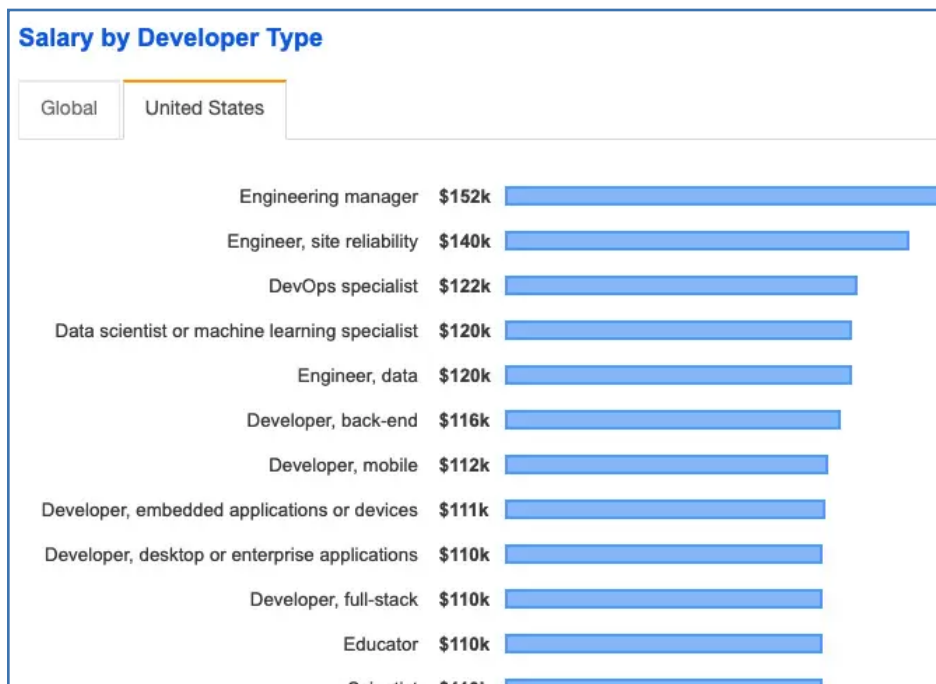
Keeping the Interview Process Positive

Evaluating a candidate involves investigating many facets of a person's skills, personality, and communication abilities. Some of your best candidates might not do too well on a skills assessment, but work very well with teams and draw out the best in each of their coworkers. If you rely completely on one facet of the hiring process, you can unknowingly turn away many of these (otherwise) excellent candidates.

> "The age-old adage 'people don't remember what you say, but they remember how you made them feel' is so true when it comes to interviews. When a candidate keeps the flow going in a friendly conversation, instead of making me feel like I am conducting an interrogation, I will remember our brief time together fondly." Refael Zikavashvili, CEO of Pramp

Part of keeping these candidates in the pool is making sure the interview process, from application to determination, is a positive experience. You don't want to take a coding result and outright reject someone with a lower than average score. After all, they might be coachable, or they might specialize in another area that you would still find valuable despite their weakness. Encourage candidates despite their results, and move on to other phases of your interview process.

Moreover, by making your application and interview process a positive experience, you can encourage candidates to work on their skills and apply again later. You can also encourage a positive reputation, and when the application process is a positive process, you'll get more referrals from the people who partake.

Deciding Compensation and Benefits

Salary by Developer Type

Global	**United States**	

Developer Type	Salary	
Engineering manager	$152k	
Engineer, site reliability	$140k	
DevOps specialist	$122k	
Data scientist or machine learning specialist	$120k	
Engineer, data	$120k	
Developer, back-end	$116k	
Developer, mobile	$112k	
Developer, embedded applications or devices	$111k	
Developer, desktop or enterprise applications	$110k	
Developer, full-stack	$110k	
Educator	$110k	

Sooner or later, you have to discuss compensation. A software engineer, in demand, will likely be comparing offers from several different companies. They'll pay attention not just to base salary, but other benefits as well.

So what might be adequate compensation? A lot goes into it.

- Average salaries. Your candidates will know roughly what people in their position get. Going back to the Stack Overflow survey above, salaries range from $38k for an academic researcher to $95k for engineering managers, but that's worldwide. In the United States, academic researchers average $95k and engineering managers reach as much as $152k on average.

- Compensating for location and re-locating. A software engineer in rural Mississippi might expect to make less than if they lived in Silicon Valley. Regional cost of living and salary adjustments can be hugely important for candidates willing to relocate, as well.

- Benefits. Health coverage, stock options, 401(k) matching; it's all part of the overall calculation.

- Time expectations. Do you expect your engineer in the office 9-5? Are they expected to work 50 or more hours per week? Are they flexible in when they can come in and leave each day? Do they get to work from home occasionally, or regularly? These can be important considerations, especially today.

Know what the averages are for your area and for companies similar to your own, and be prepared to negotiate. The best software engineers might demand a premium, but if they perform well enough on the interviews, it may be entirely worthwhile to offer it.

Reducing Your Time to Hire

Snagging the best software engineers and other developers can require a fast process from application to offer. The longer your evaluation takes, the more likely it is that another company will come in and snap up your candidates. So how can you reduce your time to hire?
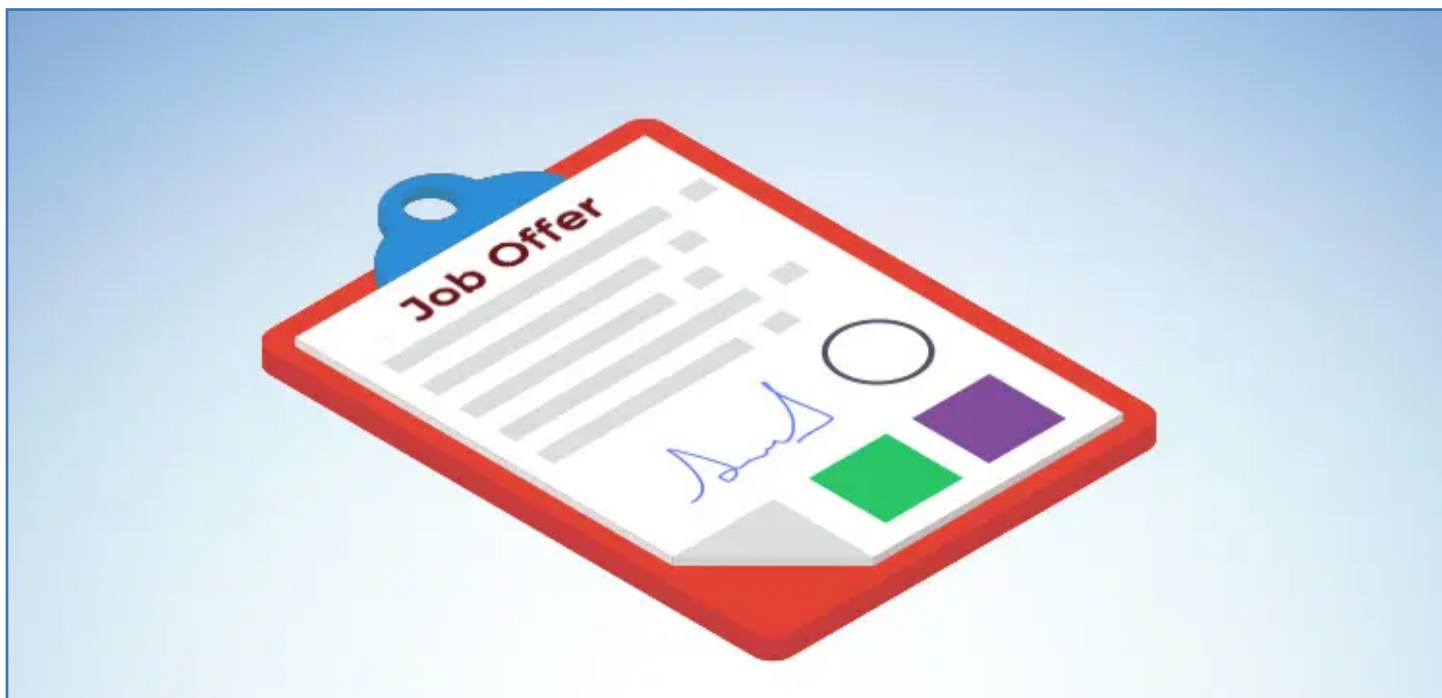
First, consider pulling the hiring managers out of the process until later. Empower a recruiter or recruiting firm to form a quality pool and identify the best candidates, and bring in your own team when those candidates have passed the initial tests and proven themselves worth your attention.

Second, make sure you have a firm structure in place for how your hiring process works. If your team is making judgments based on gut feelings and scheduling whims, you're wasting time and turning off candidates.

Third, look through your process from start to finish from the candidate's perspective. How many hurdles do they need to leap, how many roadblocks stand in their way? Can you implement tech to streamline the process? Even something as simple as a more streamlined scheduling system for interviews can save a ton of time.

Giving Them Room to Grow

Hiring your software engineer isn't the end, it's the start. You need to keep that engineer, and a lot goes into keeping a talented employee satisfied. In particular, one of the biggest drivers of turnover amongst tech employees is growth.

Specifically, you need to ensure that your developers have room to grow, from the scope of their projects to the size of their team to their compensation. If they feel like they're stagnating, they'll be more likely to look for opportunities elsewhere.

How can you provide this room to grow? In part, it means having the flexibility to increase project scope and promote talented engineers to roles suiting their ambitions. In part, it means providing incentives or reimbursement for training, certifications, and trips to industry gatherings like seminars and conferences. In part, it simply means scaling their compensation to keep up with the cost of living and other expectations.

Give your engineers a role they can grow with, and you'll have a loyal engineer who can lead your company to new heights.