

Dover

Hiring Guide: Software Engineers

Table of contents



Introduction	3
Types of engineers	4
Career Ladders	5
Hiring your founding engineer	7
Optimizing recruiting outreach	9
Designing an interview loop	11
Skills and requirements	13
Coding assessments	16
Compensation and equity	17

Introduction

You've nailed the idea. You've developed your vision. How do you bring it to life?

Bring in the engineers! It's likely that your founding team has experience in software engineering, but as you begin to scale, you'll need dedicated, advanced, full-time engineers. These are typically the first hires you make as an early-stage startup, meaning that they're not only mission critical for shipping product — they'll also set the tone for the type of company culture you'll build.

Engineering is the intersection of problem solving, strategic thinking, and technology. Engineers focus in on different areas of expertise, whether frontend, backend, or fullstack, and spend most of their time developing the product by writing code in various languages or programs.

Some day-to-day tasks may include:

- Working with Product and Design teams to develop features
- Pushing lines of code
- Resolving bugs and maintaining quality assurance
- Building feature requests for clients/customers
- Analyzing data and creating dashboards that display data in a logical way
- Managing internal tech stacks and external integrations

Types of engineers



Frontend Engineer

A software engineer who specializes in the development of the user interface (UI) is called a frontend engineer. The user interfaces include visual elements like layouts and aesthetics. Frontend engineers deal with cross browser compatibility and fixing bugs to ensure an excellent visual presentation of the UI. Thus, they work with the code that runs on different user devices, browsers, and operating systems. Developing a responsive application also falls under their responsibilities.

Backend Engineer

A software engineer who specializes in the underlying logic and performance of the application is called a backend engineer. They often design and implement the core logic, keeping in mind scalability. They do this by integrating with data systems, caches, and email systems using Application Programming Interfaces (APIs).

Fullstack Engineer

A software engineer who can handle both frontend and backend work is called a fullstack engineer. They have the skills required to create a fully functional web application.

Software Engineer in Test (QA Engineer)

A software engineer who is responsible for writing software to validate the quality of the application is called a QA engineer. QA engineers create automated tests, tools and methods to make sure that products and processes run as expected.

Site Reliability Engineer

Site reliability engineers will be dedicated full-time to creating software that improves the reliability of systems in production, fixing issues, responding to incidents and usually taking on-call responsibilities.

DevOps Engineer

Software engineers who are familiar with the technologies required for the development of systems to build, deploy, integrate and administer backend software and distributed systems are called DevOps engineers. They mostly manage the application infrastructure, i.e., the database systems, servers, etc.

Security Engineer

A software engineer who specializes in creating systems, methods, and procedures to test the security of a software system and exploit and their flaws is called a security engineer. This type of developer often works as a “white-hat” ethical hacker and attempts to penetrate systems to discover vulnerabilities.

Machine Learning (ML) Engineer

A machine learning engineer (ML engineer) is a person in IT who focuses on researching, building and designing self-running artificial intelligence (AI) systems to create and automate predictive models.

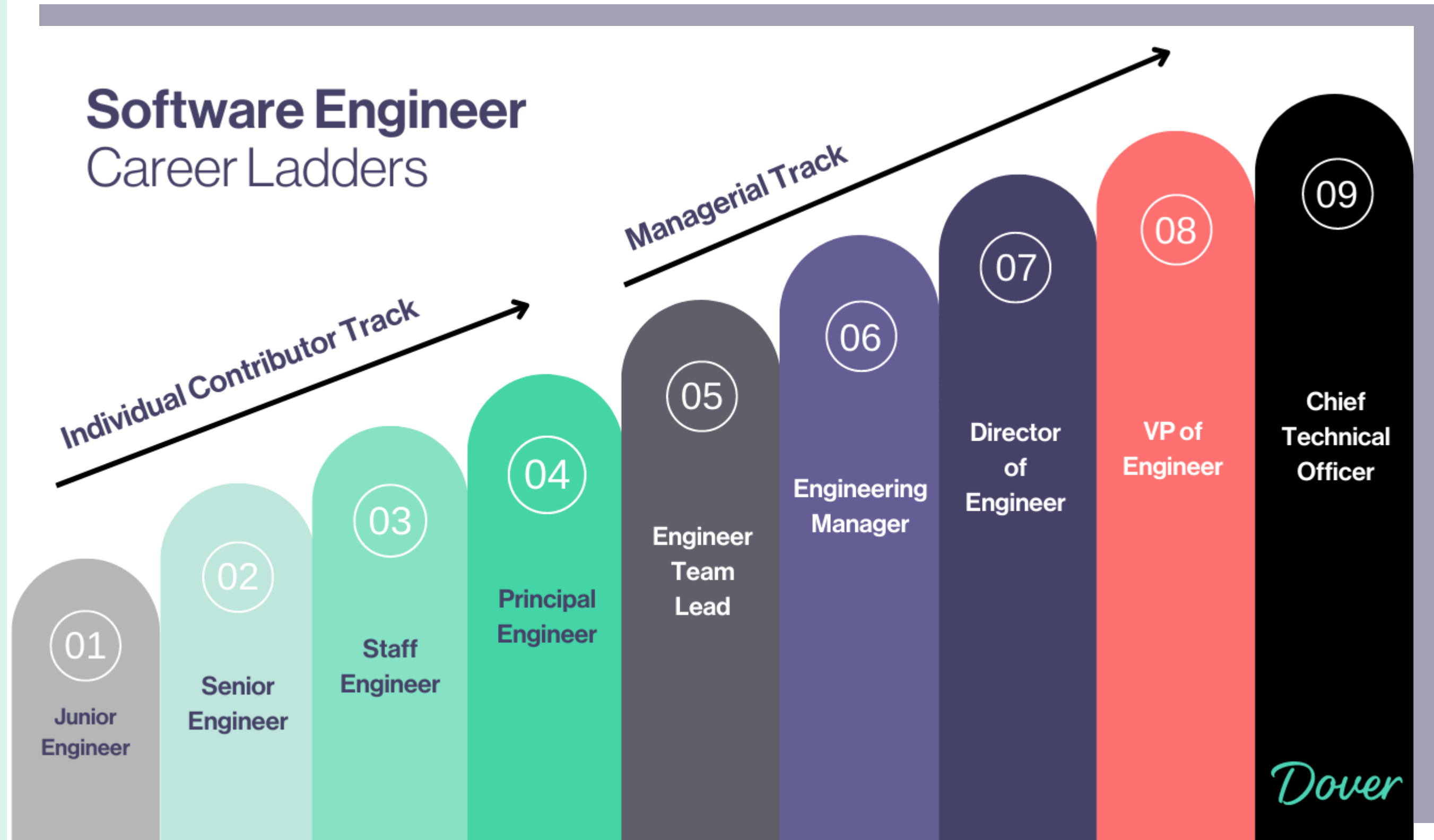
Mobile Engineer

Mobile engineer is a broad term that is used to describe anyone who builds and programs mobile applications. The most common are Android developers or iOS developers.

Career ladders for engineers

The journey from EM to Director to VP takes your role from tactical to operational to strategic. However, many software engineers prefer to code more than manage and coach a team, and they typically become principal engineers that focus on one area or department.

Here, we'll talk through the most common career path for a software engineer at a small to midsize company comparing those who want to contribute in an IC level or a managerial level.



IC

- **Junior Engineer** - These are typically newly graduated students or engineers with 0 to 5 years of experience. They handle one or more projects with guidance and direction from a senior engineer or engineering manager.
- **Senior Engineer** - Senior engineers still work in an IC capacity, but will also help train or act as mentors for entry-level engineers on their teams. They may also be deeply involved in the hiring process for junior engineers.
- **Staff Engineer** - Staff engineers still have many of the same responsibilities as a senior engineer, but whereas those were the core of their responsibilities, they're now more auxiliary. The main focus of their role is setting and editing technical direction, providing sponsorship and mentorship, injecting engineering context into organizational decisions.
- **Principal Engineer** - It's harder to pin down the day-to-day execution of this role because each person's journey depends on their own expertise, and how they can apply it to help the business achieve its goals. But they exist to guide the technical direction of the company

Engineering Management

Engineers will move through the Junior to Senior Engineer path, but instead of becoming Staff and Principal Engineers, they'll typically move into people management much sooner.

- **Engineering Team Lead** - Team leads or tech leads manage 1 to 2 ICs and are typically the most skilled engineer of their particular function. This is a rung that's skipped at many startups, as their teams aren't big enough to warrant a lead, and instead this person is promoted directly to an engineering manager.
- **Engineering Manager** - Engineering managers will typically manage one team, of around 5 to 7 people. They build a roadmap (planning and scheduling), negotiate and allocate resources (people, budgets, tools), make sure the result (tracked performance) meets specific business goals, and unblock and support their team or department.
- **Director of Engineering** - Typically where you first begin to manage managers. They're accountable for one operation area - which is why you may see titles like "Director of Engineering, Infrastructure". While Engineering Managers usually continue to write code for their team, Directors of Engineering will involve themselves much less, if at all, in committing code. Instead, their focus will be ensuring their teams are meeting deliverables and producing the correct output, coaching their reporting managers, working on the combined engineering roadmap for their area, and maximizing efficiency and collaboration.
- **VP of Engineering** - A step above the director, the VP role differs based on the size of the team and company. However, at a high level, they typically handle a larger part of the strategy and direction of the team versus the execution of a project. Directors of Engineering may spend their time figuring out how to build and maintain a significant piece of an app or product, but VPs are the ones who figure out what those pieces should be in the first place, and how they affect the company's bottom line.
- **Chief Technical Officer** - Run the entire engineering department, either by being promoted up through the management track - sometimes over the course of a whole career - or by being the founding lead engineer at a company.

pro tip:

On progression.fyi, or levels.fyi, you can compare your career ladder to those at similar or larger companies than yours to gather insights and inspiration.



Next: Hiring your first engineer

Hiring your founding engineer

Your founding or first engineer might be the toughest hire you ever make. As an early-stage startup that is pre-funding or pre-product launch, you need to enlist the help of someone from your internal network for three reasons:

1. It'll be hard to convince a stranger on your vision and work ethic
2. This person may become your future CTO — you want someone you work with well
3. Your first hires are going to take on the biggest risk (and potentially a pay cut) when joining your company — these are going to have to be folks who trust you, not just your product

Similar to investors, it's usually a bit of a red flag when the founder can't convince family, friends, or old colleagues to work with them. Though you may never get this feedback directly from candidates, knowing that you have a team who trusts in your vision and product (and enjoys working with you!) builds your credibility.



Next: Tips for your search

Top Job Boards for Engineers

General:

- [Product Hunt](#)
- [LinkedIn](#)
- [Indeed](#)
- [AngelList](#)
- [Crunchboard](#)
- [Dice](#)
- [iHireEngineering](#)
- [Engineering Central](#)
- [Stack Overflow](#)
- [Glassdoor](#)

Women:

- [Hire Tech Ladies](#)
- [In Her Sight](#)
- [Power to Fly](#)
- [Where Women Work](#)
- [Career Confessa](#)
- [Fairy Godboss](#)
- [Elpha](#)

Underrepresented groups:

- [Power to Fly](#)
- [Diversity.com](#)
- [iHispano](#)
- [Black Career Network](#)
- [Black Jobs](#)
- [HLPAs](#)
- [WorkplaceDiversity](#)
- [Diversity Jobs](#)

1. Determine what the ideal profile of your founding engineer would look like.

Not every engineering team has the same pain points, needs or tech stack. You'll want to be able to answer the following questions when considering who to source:

- 1. Will they need expertise in a specific area/tech stack?** Sourcing someone with deep knowledge of a particular platform or language is probably the route you'll want to take, as opposed to bringing on a generalist.
- 2. Will this person move up the ranks to become a CTO?** Determining how closely they'll be managing and growing your team is critical — you'll need someone with previous management experience, who has a strong network to pull from when building out your early team.
- 3. Do you want to offer a contract-to-hire option?** A contract-to-hire option offers more flexibility and limits hard feelings if the relationship isn't the right fit. Even if someone is your friend or former colleague, early startups are a fast-paced and stressful environment — even

the strongest of friendships can falter. Plus, even if the candidate ends up being a bad match, you'll still learn valuable insights into what you should look for.

2. Lean on your investors, your network and developer meetups to find the right candidate.

If you're a nontechnical founder, it may be difficult to find someone with the skills necessary to build your software from the ground up. There are a few workarounds here, but obviously, they'll require a bit more time and selling compared to someone you already know well:

- 4. Developer meetups.** Popular ATS platform Lever interviewed several founders to get their advice on hiring their first employees. Arum met her founding engineer at a developer meetup, citing that it was a positive signal to see someone invested in professional development and emerging trends.
- 5. Your investors.** It can be difficult to get candidates on the phone when you're a small team, but social proof (i.e. a personal referral) can go a long way. Your

investors have a concerted interest in helping you succeed. They likely have extensive knowledge of who is at the top of their career and might be a good fit for your team. Ask them to make an introduction on your behalf.

- 6. Your network.** Even if you don't personally know any engineers, your network certainly has folks in mind.

3. Hire engineers with design experience.

Aaron Epstein, partner at Y Combinator, recommends hiring engineers who also have design experience. Otherwise, you're left with an awesome framework and no UI or optimized user experience.

These don't necessarily need to be folks who are former designers, but you should look for those who have an eye. They'll be able to move much faster if they can iterate, code and design on their own. It also ensures you'll be creating a strong foundation that a designer won't have to come in and fix later on.

4. Have them work on an actual part of your code base.

If they pass initial interviews and screenings, invite them to work with your team on the codebase for a day.

One of Dover's co-founders, Anvisha Pai (who has been a founding engineer herself), mentions this is the best assessment of competency, and teaches you how someone will actually perform on your team outside of their credentials.

When hiring Dover's founding engineer, she gave him three potential features and told him to pick one to build out. Within three days, he had built all of them. A great thing about this process is it gives you clear conviction — the decision to hire John, one of Dover's founding engineers (who is now the CTO), was unanimous and easy.

“At Dover, we were introduced to some highly credentialed people who didn't end up being a fit and it came out pretty clearly in this process in a way that wouldn't have come up in a whiteboard interview.”



Anvisha Pai
Cofounder, Dover

Focus on the you vs. the I or we.

The goal of an initial recruiting email is solely to get candidates on the phone. You'll have an opportunity to sell them on the company during the interview process. Leave your mission statement to 1-2 sentences and spend the rest of the message focusing on the role.

- How much impact will they have?
- How much autonomy and ownership do you offer?
- How big is your engineering/design team? Who will they be working with on a day-to-day basis?
- Do you have any out-of-the-box benefits?
- What are some projects they'll work on in the first six months?
- Who will they be working alongside?

Keep sentences short, simple and concise.

Emails are read on mobile — think of the scroll test; if they have to scroll more than twice, they're probably not reading the entire email. Get the important information in first.

Use flattery and personalization.

In the crowded tech space, catching a talented candidate's eye often requires a personalized pitch. Crafting an email subject line or LinkedIn message that references one of their unique experiences or interests will get their attention much faster than a generic recruitment message.

If you can manage to establish a personal connection with a candidate, it can really pay off in the long run. **On Dover's platform, outreach that contains a personalized sentence results in 10 to 30% more interested responses** — even if those candidates have responded as not interested to another company's outreach. And even if the individual isn't looking to change careers at that exact moment, they're likely to keep you in mind once they are ready to consider new opportunities

Have a direct CTA.

Include specifics and action steps for the candidate. Writing: "Can you hop on a call within the next few days?" is much more specific and committal than "Are you free to chat sometime soon?"

Include relevant links.

- LinkedIn profile of the person sending the email
- The job posting for the role
- Your company website
- Any relevant press (if you haven't had a write-up yet, you can also consider writing a blog post about why you started your company as something to include in outreach)

Follow up 2-3 times.

Many people may not respond to your initial reach out, especially secondary connections. In fact, **our data shows that you double your response rate by sending 2-3 follow-ups at 3 day cadence.**

Your follow-ups should be shorter than your initial outreach. It's ok to reinforce a key point, but consider introducing a new concept to avoid being too repetitive. Ideas include additional press, or employee testimonials.

Optimizing outreach

30%

increase in interested responses in outreach that includes details about the candidate's interests or past roles.

Let's take a look at an example

Subject: Launch new products at BlueVine - fintech startup with \$200M raised

Hi [REDACTED],

I'm a MIT grad/engineer running a boutique search firm for software and bio startups. Your nearly 6-year tenure and rise into management at [REDACTED] caught my attention. I thought you might want to learn more about BlueVine or connect for a broader discussion of the market & your career goals. Plus, it's always good connecting with fellow alumni! :)

(And BTW, if you want to keep working in the city, I could see [REDACTED] and [REDACTED] also being worth discussing/having on your radar.)

BlueVine has raised nearly \$200M in venture funding and made over \$1B in small business loans the past 5 years. Their engineering team is based in Tel Aviv, but we're looking for a hands-on eng manager to build out a new product group in Redwood City and develop our new generation of products.

This is an opportunity to have the impact and flexibility of an early-stage startup, backed by the resources of a high-growth business. BlueVine has made a big dent in the lending world but harbors much larger ambitions. It would be great to tell you more.

Feel free to put time on my calendar for an intro call. If the timing isn't right, I'm happy to circle back later or grab a coffee. Would love to see if I can be of any help. Warm regards!

Personalization and flattery.

High-level overview of company funding, investors and customers.

Focus on impact and scope of the role.

Direct CTA — including a calendar link can double response rates.

Designing an interview loop

While hiring founding engineers will typically happen through referrals and many meetings with you and your executive team, once you have a solid foundation, you'll want to standardize your [interview process](#). At a high-level, it should contain the following stages:

Pre-interview: be transparent about the steps in your process early on.

Nerves are a part of every interview. One way to reduce stress and help your candidates put their best foot forward is to include key information about the interview process early on, in places like your job description and in outreach or calendar invites.

You should include who the engineer will speak with, as well as detail the different stages of the process.

1. Initial sell conversation

Your first intro call is an opportunity to garner interest in the role, and your company with the hopes of moving the candidate into a traditional interview process. Read more in our [Selling and Closing Guide](#).

2. Remote screen #1

The first remote screen tends to focus on the candidate's background, experiences, and any particular personality traits you're looking to evaluate. A [first-round phone screen](#) should quickly qualify the candidate for or eliminate them from further evaluation so that no one is wasting their time.

Additionally, they can help you:

- Check for potential dealbreakers early on (e.g. availability, relocation, proficiency in your tech stack, etc.)
- Ask for clarification on information in candidates' resumes or LinkedIn profiles
- Evaluate candidates' verbal communication skills (in some cases, test candidates for their ability to speak foreign languages)
- Get to know candidates
- For passive candidates, learn about their goals and desire to switch companies

Save more in-depth skills assessments for your second screening interview.

Download our [phone screen scoring rubric templates](#) to get started.

3. Remote screen #2

This second screen tends to focus on technical or practical skills.

Our key skills section will provide you with critical skills to assess during the interview loop, including examples of strong/weak responses.

This interview is important because it's the first time you're evaluating the baseline skills required to do the job. If this interview isn't properly designed, you will end up wasting everyone's time.

3. Take home assessments and work samples

Asking candidates to do part of the job helps you see and evaluate their work. The right homework assignment or exercise can even get candidates excited about the job. Homework, in particular, can also allow them to show off what they can do without the added pressure of being in an interview. However, don't overdo it. **Choose assessments that make sense to candidates**

(they should be able to understand why they're doing it and how it's relevant to the job) and try to confine their work to 2 to 3 hours at most.

4. Onsite Interview

If candidates make it to the final stage of the process, you have enough conviction that they would be able to do the job. The point of an onsite is to assess culture fit and increase the likelihood that the candidate signs.

You can read more about running an onsite interview process in [this guide](#), but at a high level, we suggest choosing 3-4 focus areas from your hiring rubric that you care the most about for a product manager. Hiring managers should assign which question set will be covered by the interviewer during each portion.

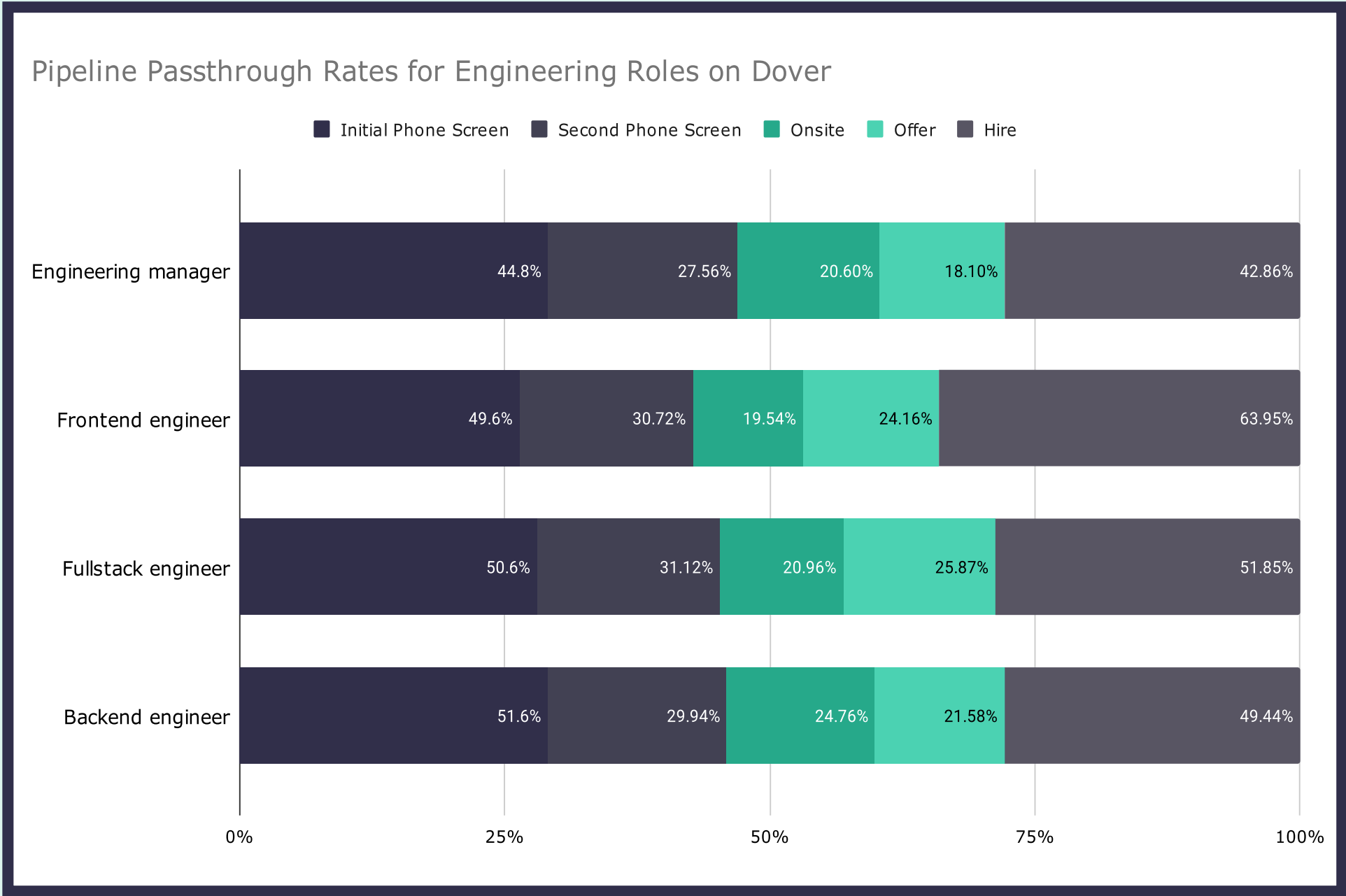
How long will it take me to hire an engineer?

In 2022, engineers are the most highly sought after role by tech startups.

The unemployment rate for tech jobs is 1.3%, its lowest level since June 2019 and about one-third of the national rate. As more startups open up their talent pool by adopting hybrid or remote work models, the projected growth rate for tech jobs over the next decade is near twice the national jobs rate. That means that hiring engineers is harder than ever.

On Dover's platform, we've seen that candidates **reach onsite 50% faster and are nearly 40% more likely to convert to an offer** when passing through our process management. We credit that to creating a positive, transparent and responsive candidate experience.

If you're curious, the graph on the left breaks this down further by engineer type.



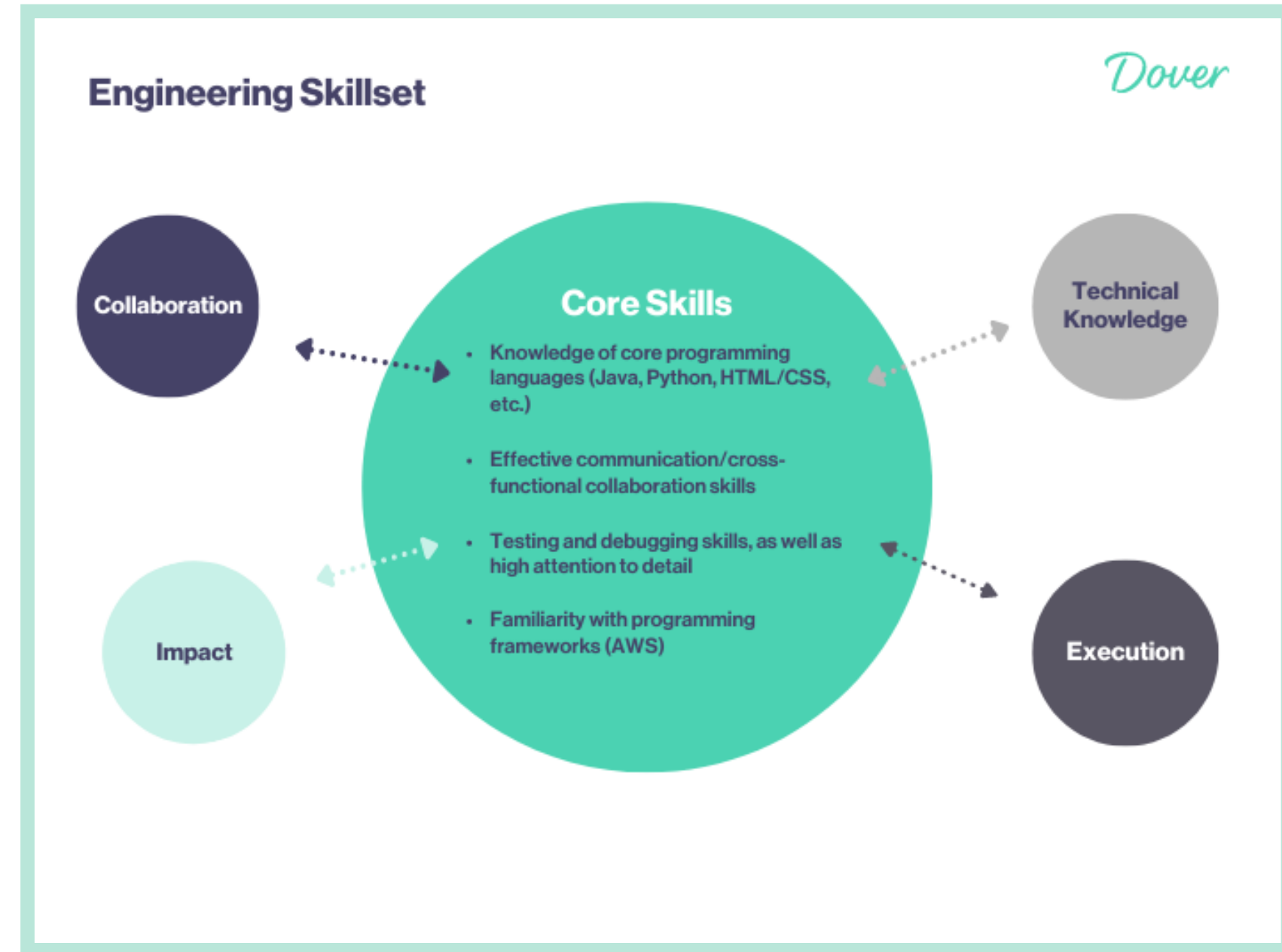
Skills and requirements

In order to be a good engineer, a candidate must have good command of technical and programming skills.

You'll also want them to have the following core skillset:

Core requirements

- Knowledge of core programming languages (Java, Python, HTML/CSS, etc.)
- Effective communication/cross-functional collaboration skills
- Testing and debugging skills, as well as high attention to detail
- Familiarity with programming frameworks (AWS)



Soft skills

Along with technical coding knowledge, the following skill categories are essential to an engineer's success. You'll want to assess these skills in the second remote screen, as well as throughout the onsite and final interview process. We've provided examples of strong and weak answers as well, so you can align everyone involved in your interview loop.

Collaboration

An engineer will have to function as a mediator between product, design and different engineering departments. Not only might they need to be good at delegating, they'll also need to effectively compromise and argue for various improvements in the product with various stakeholders. They'll also need to be able to debug, help QA other engineers' projects, and provide assistance to unblock team members if need be.

Pro tip: *Product designers have ascended to become the "third leg of the stool" alongside PMs and engineers. One way to evaluate design sensibility and product instincts is to include product designers in the interview loop.*

Example question: "Tell me about a time you disagreed with an engineer on your team and how you resolved it."

- **Weak answer:** Defensiveness, or placing blame on others, the problem resulted in gridlock or deprioritization rather than compromise and solution.
- **Strong answer:** They have humility and self-awareness. They're able to diagnose where and why they needed to compromise, and how they did so. If they end their response by saying what they learned from the situation

and how they applied these lessons going forward, they're definitely someone you're going to want on your team.

More points if they allude to/give credit to the teammates they worked with to get the project accomplished. If an engineer is using a ton of "I" or "me" statements, they might not be the best collaborator, or are poor at giving credit where credit is due.



Execution

Working at a startup or on a small team often means that engineers will receive projects without crystal clear direction.

It's not enough to have an engineer who is a talented coder — you need someone who is capable of making decisions that are in the best interest of your company's overall strategy/vision, defining deadlines and goals, as well as success metrics. Execution entails other important qualities like time-management, effective communication, and excellent people management skills.

Can you talk through a project where the product requirements or end goals weren't well defined and you had to fill in the gaps?

- **Strong answer:** They're able to discuss the roadblocks and hurdles along the development process (including how they solved them), as well as articulate how they sought feedback and revised the scope while still aligning with the company strategy or vision.
- **Weak answer:** They discuss broad strokes, or their overview is rambling, and difficult to follow.

Technical sense

To put it simply, great engineers should love to code. Especially as an early-stage company, technical knowledge is key in getting over the tough hurdles of building infrastructure from the ground-up, iterating on coding projects and design, and debugging platform issues.

How do you sanity and quality check code? What is your process for debugging an app?

- **Strong answer:** Look for a wide breadth of knowledge when it comes to writing quality code: they should mention reliability, stability, integration and testing. You should also listen for examples of humility and willingness to collaborate. You want someone self-sufficient, but modest enough to ask for help when they're stuck. Also, look for references to development and deployment testing techniques or debugging tools.
- **Weak answer:** Giving obvious or broad answers, not naming specific programs or procedures they've used in the past to QA and debug projects.

Impact

Measuring impact in previous roles is your best signal when determining the success of an engineering hire. We define some things to look out for below, but generally it's important that they're able to express what they've built in the past, and how they solved bugs, collaborated with the team, and improved on a product or feature over time.

What's the most important or impactful product you shipped? What made it so important or impactful? Would it have been as impactful without you, and why?

- **Strong answer:** Articulates tangible impact, including metrics, historical data, increases in product adoption, or revenue generation streams.
- **Weak answer:** If they only talk about qualitative impact, or don't have a good example of product innovation, leadership, or ownership. This could signal that they're not good at working autonomously.



Coding assessments

The goal of an onsite coding assessment for an engineer is to get a sense of how they approach a new problem, and how they work with others. You aren't necessarily looking to see if they get everything 100% correct — instead, you are looking at how they break down the problem, how they justify their choices and describe why they're doing what they're doing, and how they communicate it to you.

Plus, every now and then, you may encounter an engineer whose coding skills absolutely blow you away. This means that, despite job history and experience, their work far exceeds your expectations.

Here are a few more tips to ensure you're assessing candidate effectively:

Ask them to solve an actual part of your code

You can create sandbox environments to make sure you don't potentially damage or break parts of your platform, but having candidates complete real-life problems is the most accurate signal to the potential impact they could have at your company.

Include pair programming.

Pair programming is a type of screening process where your interviewer reviews each line of code as the developer writes it. This affords you the opportunity to assess how well they can walk through their decision making process, or collaborate when/if they're blocked.

Create a rubric so you can assess candidates in a standardized way.

Depending on the type of assessment you're conducting, you'll want to create a set of criteria that are dealbreakers — answers that would mean a candidate wouldn't move on to the next round. This helps to eliminate bias and assess candidates as equitably as possible.

Offer feedback immediately after the coding session is complete.

Even if they went above and beyond your expectations, this final conversation can help you understand how well they accept and incorporate feedback, and articulate their coding process.

You should also talk through how they would go through testing and deployment. Bonus points if they're able to talk about future versions of the feature, or how it would fit into the company objectives at large.



Compensation and equity



Salary

Compensation for product managers is similar to that of engineers — especially if they have coding experience. We recommend downloading our [compensation guide](#) or checking out [this blog post](#) if your startup is new to benchmarking/ researching and setting salary bands.

First, you'll want to sign up for a compensation benchmarking tool (we recommend [Pave](#)), but there are plenty of free calculators that can give you a general sense of salary data in different locations, industries, and company sizes. Regardless of which option you choose, you'll always want to cross-reference more than one tool to ensure you're making an informed decision.

In a broad sense, a solid range for junior engineers in 2022 is \$120-\$180K depending on experience, location, technical skills/coding knowledge, and the amount of scope they'll have within the engineering/product team.

Equity

Equity for startup employees is a trickier subject to tackle than compensation. It's dependent on a number of factors, including your funding, size of company and which hire number they are (i.e. the earlier the higher, the greater their risk and the higher their equity should be.)

As a general rule, you should keep you startup and equity bands within the same percentile — for example, **if you're targeting 75th percentile for equity, and the salary for that role \$155K, that would equate to 1.36% ownership in company** (for a company that has raised from \$3 to \$10 million). A great resource on equity compensation is the [Holloway Guide to Equity Compensation. Programs](#) like [Pave](#) or [OptionImpact](#) provide equity benchmarking alongside salary bands.

Free benchmarking tools:

- [Elpha](#)
- [Topstartups.io](#)
- [Angellist](#)
- [Glassdoor](#)

ROLE	SALARY RANGE
Junior Engineers	\$134K
Android Engineers	\$133K
ML Engineers	\$148K
Senior Engineers	\$175K
Principal	\$187K
Eng Manager	\$206K
Director	\$265K
VP	\$262K
CTO	\$287K

Source: BuiltIn, 2022, Median salary + equity

Dover

Looking to uplevel your recruiting strategy?

Dover is an end-to-end recruiting solution that helps high-growth teams achieve their hiring goals through automation and operational excellency.

Learn more at www.dover.com.



“Working with Dover has meant we get support on all fronts, and from every angle of recruiting including sourcing, optimal channel usage, and interview best practices.”



Sofi Pechner
Head of People Operations
Mozart Data

Helping some of the
world's best startups
recruit top talent

stripe

 **Lattice**

 **Benchling**

scale

 **INSTABASE**