How to Hire Software Engineers: The Definitive Guide

HackerRank

Win Over Top Software Engineering Talent

It's a candidate's market, and to ensure the success of your organization, you'll need to recruit better than every other business. In this guide, you'll gain insights into hiring software engineers that you won't find anywhere else.

Introduction

Realizing the world's most promising innovations — from Web3 and the metaverse to AI and self-driving cars — will require a huge workforce of talented and skilled software engineers. Yet tens of thousands of job postings sit in silence as the demand for software engineering talent grows, and the number of available software engineers fails to keep pace. Over the next decade, there will be an average of 189,200 openings for software engineers and developers each year.

And that trend is accelerating. From 2020 to 2030, the number of employed software engineers and developers in the U.S. is projected to grow by 22 percent — almost triple the 8 percent average growth rate for all occupations.

Across industries, the shift to remote-first work and virtual interviewing has further transformed technical hiring. That shift has created massive opportunities for employers. Recruiters can source talent from anywhere in the world, and hiring managers can build more distributed and diverse teams than ever before.

But even as remote hiring booms, that opportunity comes at a cost. The distance that now unites talent and teams can also pose increasingly challenging without the right kinds of processes and tools in place.

As a technical recruiter or talent acquisition professional, solving this monumental problem is the key to success. It's not hyperbolic to say that the success of an organization hinges on its talent acquisition team's ability to build scalable, efficient, and high-performing hiring processes to land the most sought-after talent in the world.

So, where does one even begin?

To win over the best software engineering talent, companies will need to recruit better than every other business out there. They need to cultivate stellar candidate experiences and iron out efficient and effective processes. They need to hone their storytelling skills to attract, close, and retain software engineers. And they need to develop fluency over an increasingly complex developer skills landscape to sell the technical requirements of the opportunity.

In this hiring guide, we'll break down everything you need to know about recruiting software engineers. From sourcing to closing candidates, you'll gain key insights into hiring for this role that you won't find anywhere else.

Your Blueprint for Hiring Top Software Engineers

1. **Software Engineering 101**
2. **Meet a Software Engineer:** Role Demographics
3. **Plan:** Build an Efficient Hiring Process
4. **Source:** Find Qualified, Diverse Candidates
5. **Screen:** Evaluate Top Engineers, Faster
6. **Interview:** Assess Real-World Technical Skills
7. **Rank:** Close the Role With the Right Hire

1 Software Engineering 101

Software engineering is a branch of computer science that deals with the design, implementation, and maintenance of complex computer programs. Software engineers develop, design, and test software applications and apply engineering principles to the process of building software.

Core Responsibilities

A software engineer performs many of the tasks that a software developer does, but with a big-picture, design-centric approach. This distinction means that they're focused on structure design and the elimination of technical debt, in addition to writing great code.

On a more technical level, the core job responsibilities of software engineers include:

- Writing high-quality, maintainable, reusable code
- Collaborating with and demonstrating features to stakeholders in an Agile environment
- Designing, developing, troubleshooting, and debugging software programs for enhancements and new products
- Conducting routine concept design through the full development life cycle
- Preparing and conducting systems programming tests

Key Terminology

Software engineering is an advanced discipline with its own set of technologies, terms, and jargon.

Talent acquisition teams and hiring managers who have fluency over technical language will have a competitive advantage while recruiting for roles.

Some core technical concepts include:

Agile: An iterative approach to project management and software development.

AI: Artificial intelligence. At its most basic, that means the ability of a digital computer to perform tasks associated with intelligent beings.

Algorithm: A set of rules followed in calculations or problem-solving operations.

API: Application program ittnterface. A connection that computers and applications use to communicate.

Container: A standard unit of software that packages up code and all its dependencies.

Framework: A platform that provides a foundation for developing software applications.

Language: A language used to control the actions and behavior of a computer.

ML: Machine learning. The development of computer systems that are able to learn and adapt without following explicit instructions.

Object-oriented programming: A technology for writing programs that are made up of self-sufficient modules that contain all of the information needed to manipulate a given data structure.

Open source software: Software for which the original source code is made freely available and may be redistributed and modified.

Relational database: A database structured to recognize relations among stored items of information.

Key Technologies and Languages

C A general-purpose programming language that is structured, compiled, and statically typed.	**C++** A general purpose programming and compiled language created as an extension of the C programming language, or "C with Classes".	**C#** A general purpose, object-oriented programming language developed around 2000 by Microsoft as part of its .NET initiative.	**Go** A popular language invented at Google in 2007. Originally built for networking and infrastructure, Go is now a general purpose language used in a wide range of applications.	**Java** A high-level programming language used to create complete applications.
.NET An open-source cross-platform framework and successor to Microsoft's .NET Framework. Enables developers to use multiple languages, editors, and libraries to create applications.	**PHP** A widely-used open source general-purpose scripting language that is especially suited for web development.	**Python** An interpreted, high-level, general-purpose programming language, and one of the most popular languages for rapid development across multiple platforms.	**Ruby** An interpreted, dynamic, open-source programming language with a focus on simplicity and productivity.	**SQL** Structured Query Language. An industry-standard query language that works with relational databases.

Due to the techonlogy industry's pace of innovation, this list of software engineering technologies is constantly evolving. The popularity of coding languages changes every year and new tech is constantly being developed. Research into developer skills will be necessary to understand the unique technical requirements — and opportunities — for each role.

Skills and Competencies

Familiarity with engineering vernacular and technologies is important in building fluency. Once that fluency is developed, you can hone the way in which you think about assessing talent and determining their skill levels and competencies — and even more critically, communicating the technical opportunity of the role at hand.

In assessing key skills and competencies, they're often broken down into the following categories:

Code Quality

When multiple developers work on the same codebase, it is important for them to follow best practices to make sure unintentional patterns and syntax errors do not get committed. That is often referred to as **code quality.** This competency area includes understanding the benefits of writing code that is modular, reusable, maintainable, and secure.

Key Competencies:

1. Modular code - Includes writing code that has both independent modules and better code organization.
2. Reusable, maintainable code - Includes writing code that is extendable to new applications and programs.
3. Secure code - Guards against the accidental introduction of security vulnerabilities.

Language Proficiency

When peers in a technical role talk about a candidate's **language proficiency**, they're referring to the candidate's ability to understand all of the rules, features, and mechanisms of a programming language, and optimize for simplicity, accuracy, and modularity within the rules of the programming language to implement solutions.

Key Competencies:

1. Writing code - Ability to write code in a given language and proficiency in using common language features. Ability to produce a functional program with ease.
2. Optimizing code - Optimizing source code of a given language.

Skills and Competencies

Technical Communication

Technical communication is used to make information about technical products, services, and processes clear and understandable. It is the ability to convey thoughts, designs, ideas, and specifications in a clear and concise manner.

Key competencies:

1. Communicate technical concepts .
2. Communicate technical knowledge with peers.

Problem Solving

Solving problems is the foundation of computer science. At its core, **problem solving** focuses on the study, understanding, and usage of data structures and algorithms.

Basic competencies include:

1. Basic data structures - Use data structures such as arrays and strings. Traverse through arrays, strings, and linked lists. Access and update individual elements in array, and characters in strings.
2. Basic algorithms - Create simple sorting algorithms such as bubble sort, merge sort, and counting sort. Create simple brute force and sub-optimal solutions.

Intermediate competencies include:

1. Data structures - Use data structures such as HashMaps, stacks, queues, and heaps. Manipulate single or double linked lists.
2. Algorithms - Use optimal solutions to solve problems. Analyze run-time complexities and space complexities. Create simple dynamic programming-based algorithms.

Advanced competencies include:

1. Advanced data structures - Work with data structures such as trees or graph traversal. Find the shortest paths, diameter, etc. Use trees such as binary search effectively.
2. Advanced algorithms - Use dynamic programming and specialized algorithms. Knowledge of efficient string search algorithms such as Rabin-Karp and Knuth-Morris-Pratt. Efficient graph search algorithms such as Dijk.

Experience and Education

After skill competencies, the most important qualification for software engineers is experience. For some employers, on-the-job experience and training is a critical requirement.

Then, there's the question of education. About 75 percent of developers worldwide have a bachelor's degree or higher. If you're recruiting, screening, or hiring engineers, there's a high likelihood that many of them will have a degree. And many companies still require engineers to hold four-year degrees.

But companies looking to hire software engineers should also be prepared to recognize other forms of education and experience. Competition for skilled software engineers is sky-high, and it's not uncommon for job openings requiring a degree to go unfilled. But employers that prioritize real-world skills over pedigree gain access to a much larger pool of skilled talent.

Research has shown that 86.7 percent of developers have taught themselves a language, framework or tool outside of a classroom or course. And bootcamps and online training are popular ways to learn new technical skills.

It's becoming increasingly more common for the best candidate for an open role to be one that has no higher education or on-the-job experience. Because of this, many of the world's leading tech companies no longer require a four-year degree. Employers that are willing to hire engineers from a diverse range of backgrounds will have a much easier time scaling their team.

Responsibilities by Seniority Level

ENTRY-LEVEL

Years of experience: 0-3

Responsibilities: execution of solutions, testing, debugging, quality assurance

MID-LEVEL

Years of experience: 3-5

Responsibilities: solving technical problems with unknown solutions

SENIOR-LEVEL

Years of experience: 5+

Responsibilities: solving complex and ambiguous problems, understanding business-level implications

The titles a software engineer may hold vary drastically, depending on their experience and the company or industry in which they work. The title of a graduate from a coding bootcamp might look different than a candidate with a bachelor's degree. And the role of an engineer in a five-person startup will differ from one at a 5,000-person company. The expectations and responsibilities of the engineer also scale based on experience.

At the beginning of their career, a software engineer will typically start with an entry-level title like Software Engineer 1. A new engineer will spend a lot of time on testing and quality assurance while learning the internal systems. The tasks they are given are framed around execution of solutions. They'll typically work in that role for one to three years.

After gaining experience in an entry-level role, engineers typically have the opportunity to move into mid-level roles. At this stage, the parameters of their work become more ambiguous. Instead of basic execution, they're now responsible for solving technical problems with unknown solutions.

From there, they may have the opportunity to move into more senior-level roles with hands-on development and engineering responsibilities, such as senior software engineer, lead software engineer, and senior software architect. How long it takes to reach this level depends on the company and the skill level of the engineer.

The key requirement to reach senior-level engineering roles is the ability to handle even more complex and ambiguous problems, with an uderstanding of their implications on the business. At this stage in their career, the work that senior engineers do becomes exceptionally valuable to the teams that hire them.

While they spend several years honing these skills, their responsibilities often expand to include taking more ownership of projects, working independently in a team environment, and mentoring team members. Senior engineers often specialize in particular technologies, such as databases, cloud computing, information operations, or systems architecture.

Sample Job Description

The job descriptions for software engineering roles can vary widely, depending on the responsibilities, compensation, and seniority of the position. That said, there are commonalities between descriptions that you can take advantage of. Here's an example of a job description for a mid-level software engineering role.

Title: Software Engineer II
Full-time. Associate.

Responsibilities

Engineering - Responsible for design, development, testing, deployment, and support of secure and scalable enterprise solutions.

Delivery - Agile delivery of solutions aligned to business needs while maintaining a high standard of quality. Participate in code reviews and troubleshooting of live site issues and bugs.

Collaboration - Partner with product owners to understand business and product requirements to translate them into engineering solutions.

Communication & planning - Clearly communicate roadmap, backlog, and team updates across the organization.

Qualifications

Basic Qualifications

- Programming experience with at least two software programming languages.
- 3-5 years of experience in software development.
- BS/BA degree or equivalent experience.

Required Qualifications

- 3+ years of hands-on development experience with a combination of Java, Angular, Python, or React JS.
- 3+ years of hands-on SQL development experience.
- Advanced knowledge of application, data, and infrastructure architecture disciplines.
- Understanding of architecture and design across all systems.
- Knowledge of industry-wide technology trends and best practices.

Other Desired Skills

- Full-stack development experience.
- AWS frameworks and coding (Lambda, SNS, Cognito).
- DevOps/CI/CD tools experience.

Salary Information

On average, software engineers tend to receive a salary higher than the national average for all occupations in their country of origin.

In the U.S., for example, the average salary for all workers in 2020 was $53,400. In contrast, the average base salary for software engineers in the U.S. is $117,993 — 121 percent higher than the national average. What's more: that number can differ drastically based on region and industry.

Entry-level software engineers can expect to occupy a lower salary band at the beginning of their career. In contrast, more senior positions provide a higher average compensation. Software engineering salaries vary widely based on a number of factors, including industry, location, and company size.

Market conditions such as labor shortages and the Great Resignation have made technical salaries especially volatile, and many salary statistics are quickly becoming out of date. Additional benefits, including equity, bonuses, and stipends, are also changing rapidly — and can be used to increase the monetary value of an overall offer.

Note: Recruiters will need to conduct independant research to identify salary bands based on their company's requirements, current market conditions, and the technical needs of the role.

Software Engineer Salaries in the United States

Years of experience	Annual Compensation
<1 year	$110,255
1 to 2 years	$113,276
3 to 5 years	$115.515
6 to 9 years	$129,565
10+ years	$148,437

(Sources: Software Engineer Salaries in the U.S. Engineering Lead Salary in United States)

2 Meet a Software Engineer: Role Demographics

Meet a Software Engineer

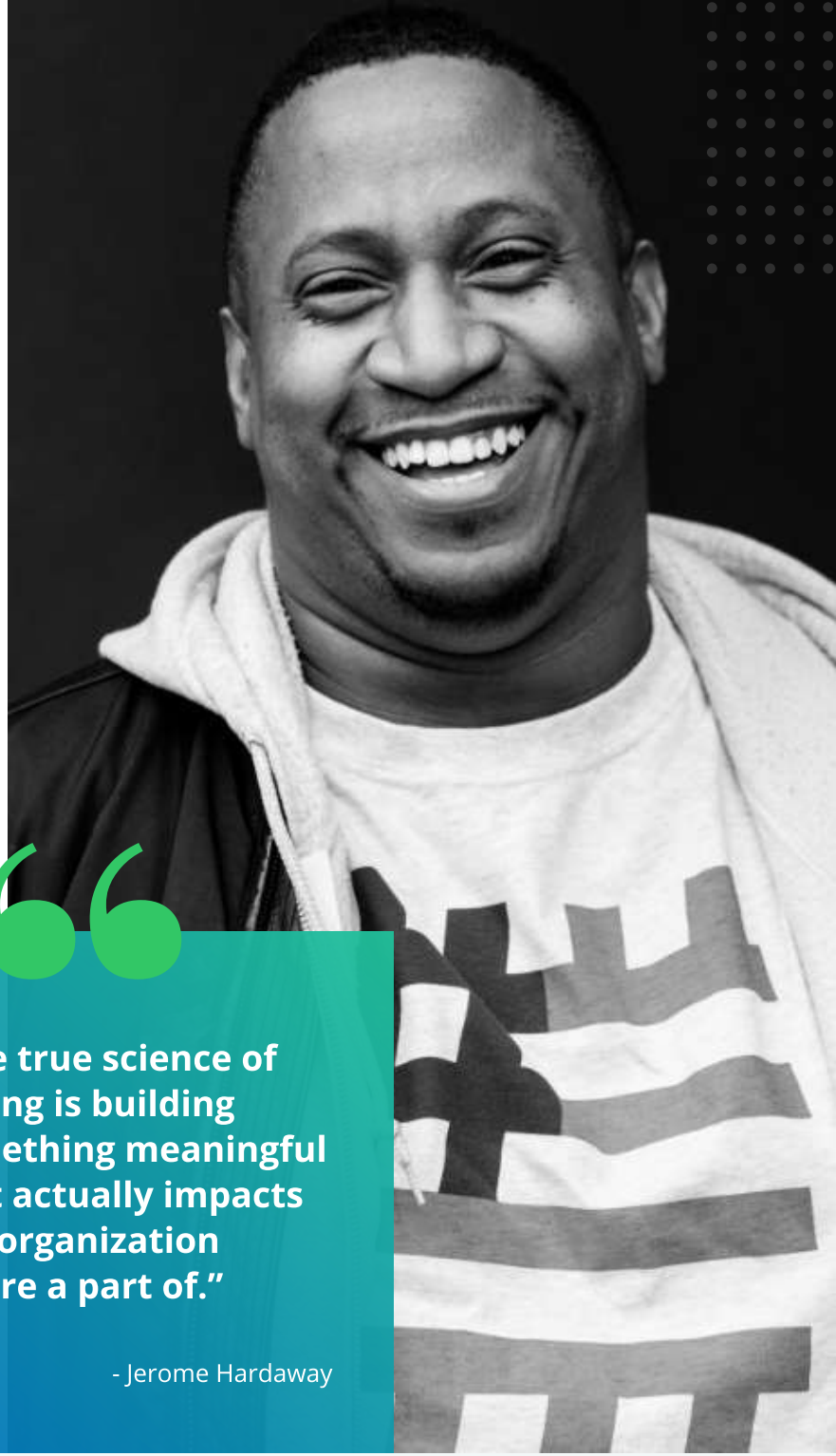
From Air Force veteran to Microsoft engineer, Jerome Hardaway has been on an arduous but rewarding journey.

Jerome spent five years in the United States Air Force before leaving in 2009. But the decision to become a software engineer wasn't a natural progression. During the Great Recession, he decided to transition to software development because it offered more employment opportunities.

Jerome found the transition from the military to tech to be surprisingly smooth. "Looking at firearms and writing code, they are really similar," said Hardaway. "When you compare the memorization requirements, they correlate perfectly."

In 2014, Jerome founded Vets Who Code, a non-profit dedicated to filling the nation's technical skills gap with a training program for veterans. As the non-profit's executive director, Jerome has seen first hand how transferable skills and a diverse background can enable vets — and any candidate with a unique career path — to thrive on technical teams.

Throughout his career, Jerome has worked on engineering teams at a number of well-known companies, including CBS Interactive, Quicken Loans, and Microsoft. He's also won several honors, such as being named a Twilio Champion, GitHub Star, and Google Developer Expert.



"The true science of coding is building something meaningful that actually impacts the organization you're a part of."

- Jerome Hardaway

Role Demographics

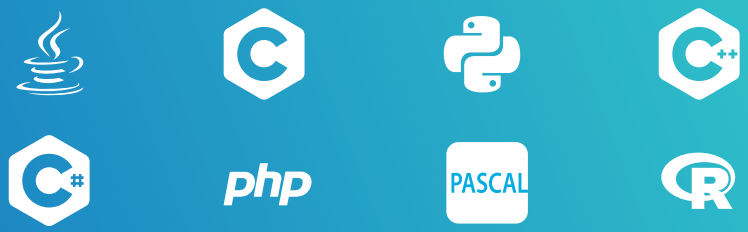
Jerome's experience isn't unique. Today's software engineers come from backgrounds that span a wide range of experiences and demographics.

Research has shown that 86.7 percent of developers have taught themselves a language, framework or tool outside of a classroom or course. And the number of developers graduating from bootcamps increased 1,046 percent from 2013 to 2020.

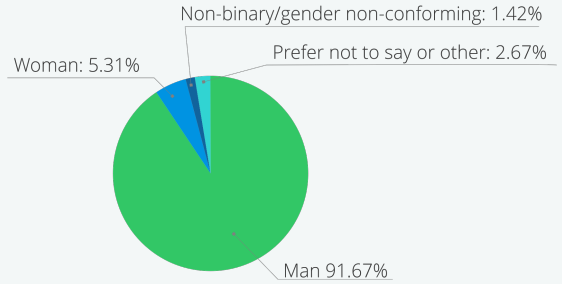
24.5 M
Number of Software Engineers Globally

39.5
Average Age of Software Engineers in the US

Top Programming Languages for Software Engineering

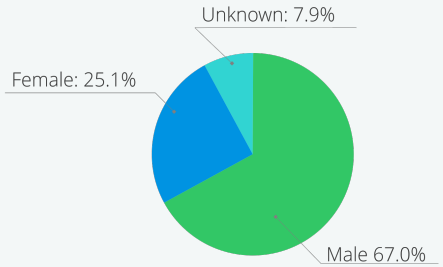


Software Developer Gender Distribution Worldwide in 2021



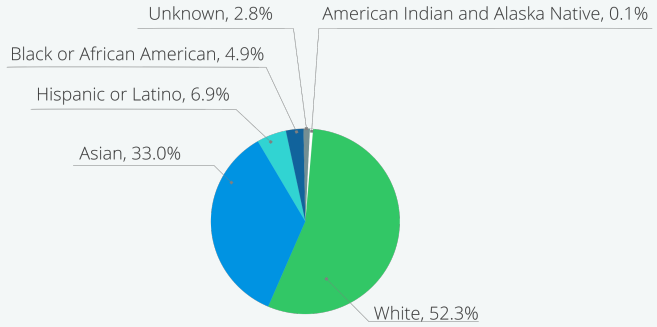
(Source: https://www.statista.com/statistics/1126823/worldwide-developer-gender/)

Software Engineer Gender Distribution in the US:



(Source: https://www.zippia.com/software-engineer-jobs/demographics/)

Software Engineers by Race in the US:



(Source: https://www.zippia.com/software-engineer-jobs/demographics/)

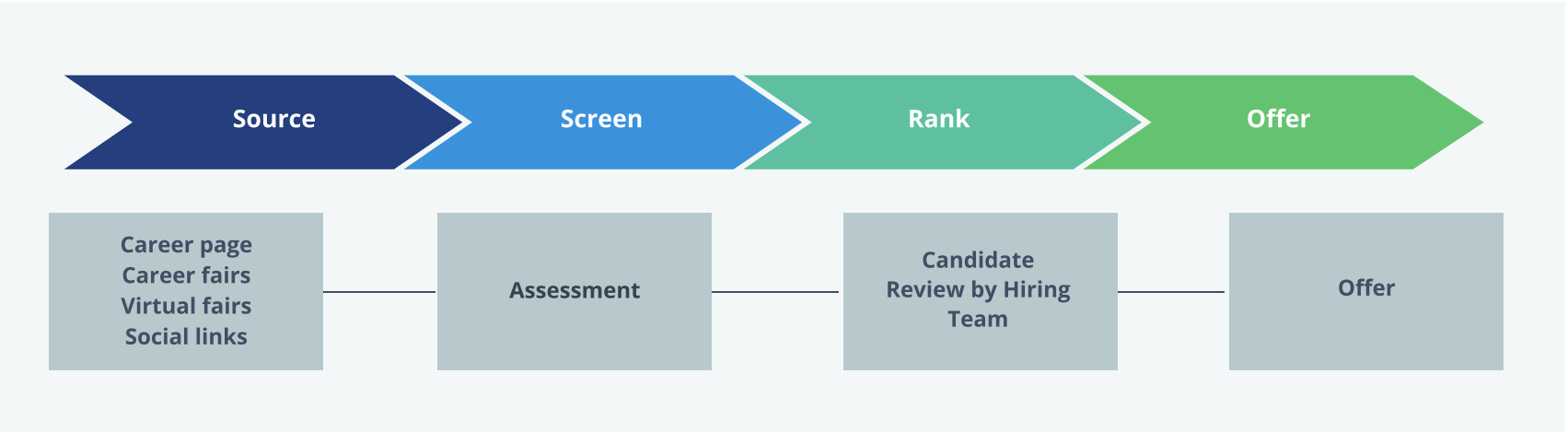
3 **Plan:** Build an Efficient Hiring Process

The challenges of acquiring technical talent depend on the seniority level of the talent. Recruiting a senior engineer, for example, is a distinct challenge from recruiting college graduates. As such, an effective hiring process is one that's tailored to the seniority level of the individual role.

Hiring Process for Entry-Level Engineers

Hiring early-talent engineers is vital to a company's success. Not only do they help shape the company's culture, but as they develop skills, they help drive the future development of the company's road map. Early talent usually consists of recent college graduates or engineers with less than three years of hands-on, real-world experience.

Below is a recommended process for hiring early-talent engineers. Because early talent tends to involve a higher volume of applications than other experience levels, the hiring process will focus primarily on sourcing and screening applicants.



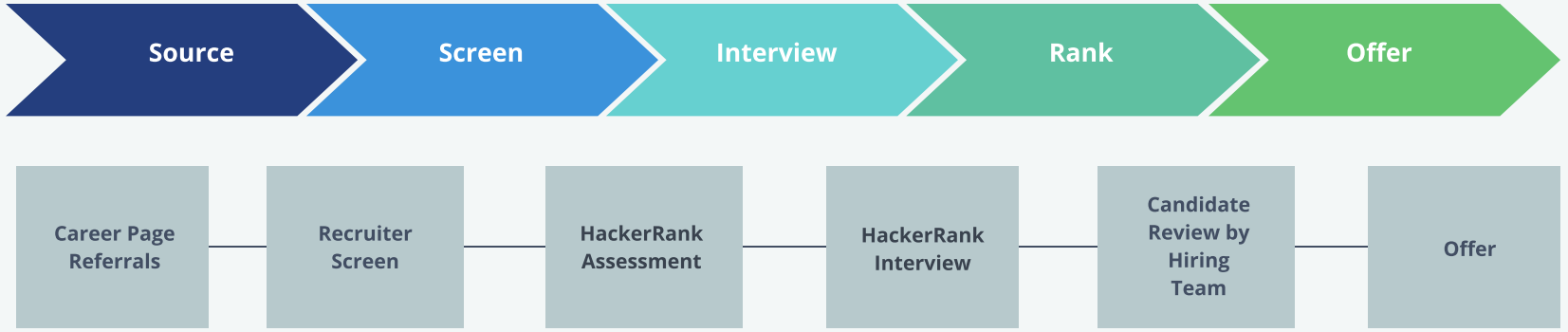
Tailor the Approach

Hiring Process for Mid-Level Engineers
Mid-level engineers, who have three to five years of experience, are one of the most in-demand experience levels. The need for mid-level talent is accelerating, but the supply of skilled engineers is limited.

Because mid-level engineers are well into their career, one of the main sourcing channels that recruiters have at their disposal, university recruiting, is no longer available for mid-level roles. To compensate for this gap, talent acquisition teams must rely on a number of proactive sourcing channels, including networking, referrals, virtual events, and social media.

In addition to proactive sourcing, companies with aggressive hiring goals will need to build a world-class candidate experience that attracts and converts talented engineers at scale.

The below visual shows a standard hiring process for a mid-level role. The key difference from the entry-level process is that there is often a phone screen with a recruiter early in the process and the addition of project-based assessments in the form of pair programming or virtual whiteboards.



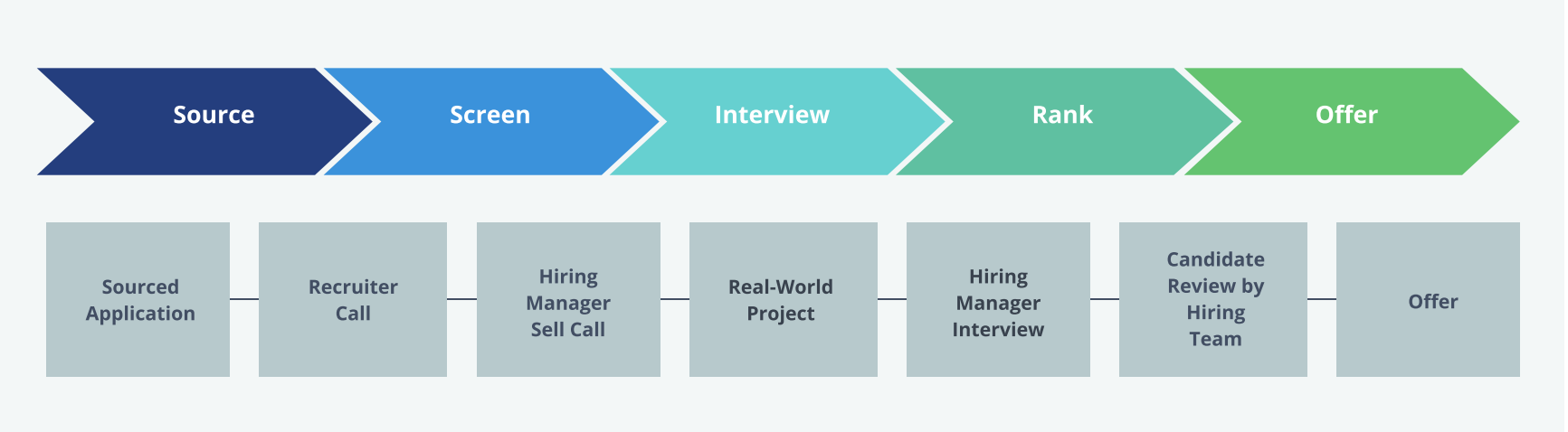
Customize for Experience

Hiring Process for Senior Engineers
Hiring senior talent is a much different process than the one used for entry-level and mid-level talent.

The competition for senior-level talent is fierce. Many senior engineers are already employed, and you will find that there are fewer candidates on the market. This is because software engineers as a whole tend to be more passive in their job searches, with this being particularly true for senior-level candidates.

A study by LinkedIn found that U.S. software engineers are 13% less likely to apply to a job opportunity than the global population, but 12% more likely to respond to a message from a recruiter. This means that software engineers are receiving more inquiries and offers, while the companies recruiting them have a shortage of applicants. To reach candidates who are both scarce in supply and passive in their search, talent acquisition teams must rely on strategic outreach and internal referrals to meet aggressive hiring needs.

Hiring managers will also want to conduct an early call with the candidate to sell the opportunity and generate enough interest for the engineer to want to move forward with the conversation.



4 **Source:** Find Qualified, Diverse Candidates

Employer Branding
When hiring entry-level and mid-level professional engineers, a main challenge of talent acquisition is building a strong, passive, inbound sourcing pipeline that attracts candidates to the company. And the key to building that pipeline is a strong employer brand.

Industry leaders and startups alike have proven that the best way to attract early- to mid-career talent is by building a place where engineers want to work.

To see the power of employer branding in action, you only need to look at hiring data for the world's largest tech companies. Google, for example, receives over 3 million applications every year because of its reputation among potential candidates.

Recruiters looking to source great candidates should focus on this passive channel, and complement it with additional active sourcing channels, such as university hiring and networking.

University Recruiting
Tens of thousands of students graduate every year with degrees in software engineering and computer science. Fewer than 50,000 students graduated with a computer science degree in 2017. And there were only 3,365 computer software engineering degrees in 2019. (Both are common degrees among software engineers.)

Campus recruitment offers talent acquisition professionals the opportunity to create a long-term talent pipeline with new software engineers and build an employer brand affinity with new software engineers.

But as important as university recruitment is for finding great engineering talent, it can also prove challenging.

The seasonality of university hiring makes it one of the most dynamic arenas of recruiting. Without constant iteration and re-strategizing, it's easy to fall behind the pack. And especially for technical candidates, competition is stiff. Having a technical university recruiting program isn't enough—companies need world-class programs to stay competitive.

Taking an analytical approach is vital to creating a more effective university recruiting program. Start by identifying your hiring objectives and program goals. Then identify key performance indicators (KPIs) that intersect with the two goals.

Identify University Hiring KPIs

Try this chart as a jumping off point. Take a look at the X axis (business goals) and the Y axis (program goals) to identify your top 3 goals for each category. From there, you can identify the KPIs most relevant to your goals at the box where the two intersect.

		Business goals						
		Decrease time to offer	Increase application completion rates	Increase offer accep-tance rate	Maintain positive candidate experience	Recruiter time savings	Hiring manager time savings	Retention
Program goals	Maintain candidate engagement	Time of offer for engaged candidates vs. cold candidates	Outreach: application fill rate	# Touches per role filled	Candidate feedback: overall experience	Recruiter time spent on engagement	Hiring manager time spent on engagement	Length of pre-hire engagement vs. hires retained
	Minimize candidate drop-off	Late stage candidate drop-off rate	% incomplete applications	Rejected offer rate vs. final stage interview drop-off	Hightest drop-off stage	% Drop-off after recruiter screening stage	Drop-off after HM interview stage	NPS change over time (on avg, by candidate)
	Refine skill set targeting	Time to fill per role	Filled applications per role	Average offer acceptance	Candidate feedback: Role-based	Recruiter ease of participation (rating scale)	# Candidates sent to HM interview stage per role	# Candidates retained per role
	Increase event ROI	Time to fill by event	Event contacts gained vs. applications completed	Event contacts met vs. offers accepted	Post-event student feedback	# Candidates sourced per event	# Candidates sent to HM interview stage per event	# Candidates retained by event type
	Increase hire quality	Time to assess technical skills	Average assessment score vs. # of compiled applications	# Offer acceptances from top X% of candidates	Feedback: candidates with high assessment scores(NPS)	Total hours spent sourcing vs. # of passed assessments	Total hours spent verifying tech skills per candidate	Attrition analysis: hires let go vs. hires departed

Expand Your Talent Pool

Another key is to expand your talent pool beyond prestigious, four-year colleges. The competition for top talent is fierce, with the demand for technical talent far exceeding the number of graduates. To fulfill aggressive entry-level hiring goals, you'll need to look for talent in new places you wouldn't expect.

You could, for example, explore less traditional "top university" lists to expand your search:

- Best Colleges for Veterans
- Colleges with High Economic Diversity
- Most Diverse Campuses
- Colleges with the Most Students Over Age 25

And if you have the latitude to branch outside traditional universities, consider including options like coding bootcamps, high school programs, and community colleges, too. Their rules of engagement differ from traditional universities, but still boast a high volume of eager junior candidates.

Sourcing Senior Software Engineers

Hiring senior software engineers is a fundamentally different challenge than hiring entry- and mid-level engineers. The demand for senior, technical talent far exceeds the potential pool of senior engineers.

Few companies get enough passive applications for senior engineers. If your team needs to hire a number of senior engineers, you'll need to turn to outbound sourcing. One strong recommendation is to base your approach off of the principles of **account-based marketing**. In this model, talent acquisition works to identify prospective candidates who aren't on the job market and reach out through networks and social channels.

The key, though, is that the outreach should be highly compelling — and better still, personalized. Assume the candidate already earns a lofty salary and will receive dozens of similar inquiries. To catch their interest, you'll need to anticipate their career goals and motivations, and develop a storytelling strategy to encourage engagement.

Source Diverse Candidates

If your company is committed to diversity, equity, and inclusion, you should evaluate your job description language and leverage diversity sourcing tools to attract diverse talent.

Job Description Language

A 2018 Muse User Survey found that 55% of candidates consider job descriptions as one of the most important factors when deciding if a company's a good fit. That means the language you use to construct a job description will influence a candidate's opinion about the company. Talent acquisition teams committed to diversity, equity, and inclusion should pay careful attention to how candidates from a variety of backgrounds will interpret the job description.

An inclusive job description speaks to diverse applicants while being specific about the skill sets required. Leading with inclusive language shows candidates you're an inclusive workplace that considers all applicants regardless of their identity. For example, using gender-neutral words in job listings helps attract applicants to roles they might not otherwise apply for.

Sourcing Tools

Sourcing talent from standard recruiting channels — think LinkedIn job postings and company career pages — might not be enough to attract diverse talent. In contrast, starting a search with tools designed to reach underrepresented communities will create a more diverse pool of applicants. It will also provide access to talent that the competition won't know about. Below are some examples of diversity sourcing tools that will prove useful during a candidate search.

- Ability Links
- Black Career Network
- Boolean Search Strings for LinkedIn
- Entelo Diversity Tool
- Fairygodboss
- The Mom Project
- Vet Jobs

5 **Screen:** Evaluate Top Engineers, Faster

Once you've built a strong sourcing funnel, a new challenge will arise. How do you screen a high volume of applications to identify the best candidates? In this section, we'll break down key components of the screening stage, with a focus on assessing software engineering skills to find the right talent for your business needs.

Resume Screening

Traditionally, companies don't ask coding interview questions until after a candidate has gone through resume screening. But when facing a flood of candidates for a role, it can be difficult to gauge their skills based on resumes alone. A software engineering resume doesn't adequately communicate a candidate's abilities, such as deep technical skills, code quality, and language mastery.

Resumes are an important part of hiring technical talent. But when resume screens are administered, and to what degree they're emphasized, can vary depending on the needs of the organization.

Most technical teams looking to hire the best candidates supplement the resume screen with a skills assessment to gauge the candidate's real-world skills and ability to do the job at hand.

DEI Best Practice: Screen Out PII

Many hiring managers use a resume-first process to begin assessing candidates.

The problem is that resumes often contain a candidate's personally identifiable information (PII), like name, email, school, employment history, and more. Hiring managers may unconsciously use this information to assume demographic information, including age, race, gender, ethnicity, and nationality.

The result is that hiring managers may inadvertently introduce unconscious bias into the hiring process.

Screening out PII allows hiring managers to review the candidate's work history before seeing any personal or demographic information. This helps the hiring manager's focus on the application without unconscious bias can get in the way.

When using any resume screening or skill assessment tool, it's important to look for options that allow hiring teams to screen out PII.

Phone Screens

Best Practices and Sample Questions
Phone screens are quick-fire ways to evaluate communication skills and weed out candidates who don't meet key requirements for the role. The phone screen is often a brief conversation between the recruiter and candidate to establish basic qualifications and develop rapport.

During the call, questions about a candidate's previous role and their knowledge about the company work well. Here are a few examples to get started:

- How did you hear about our company and this role?
- What tasks did you perform on a daily basis at your previous company?
- Tell me about a career goal you have.
- Why do you want this job?
- What management style do you prefer?

Beyond these basic questions, some organizations may ask technical problems, such as object-oriented programming questions or high-level design of a small system.

Another critical, but often overlooked, component of the phone screen is early communication of company messaging and the employer value proposition. The more you sell the candidate on the role during this stage, the better the candidate experience will be, and the higher the acceptance rate will be.

Skills Assessments

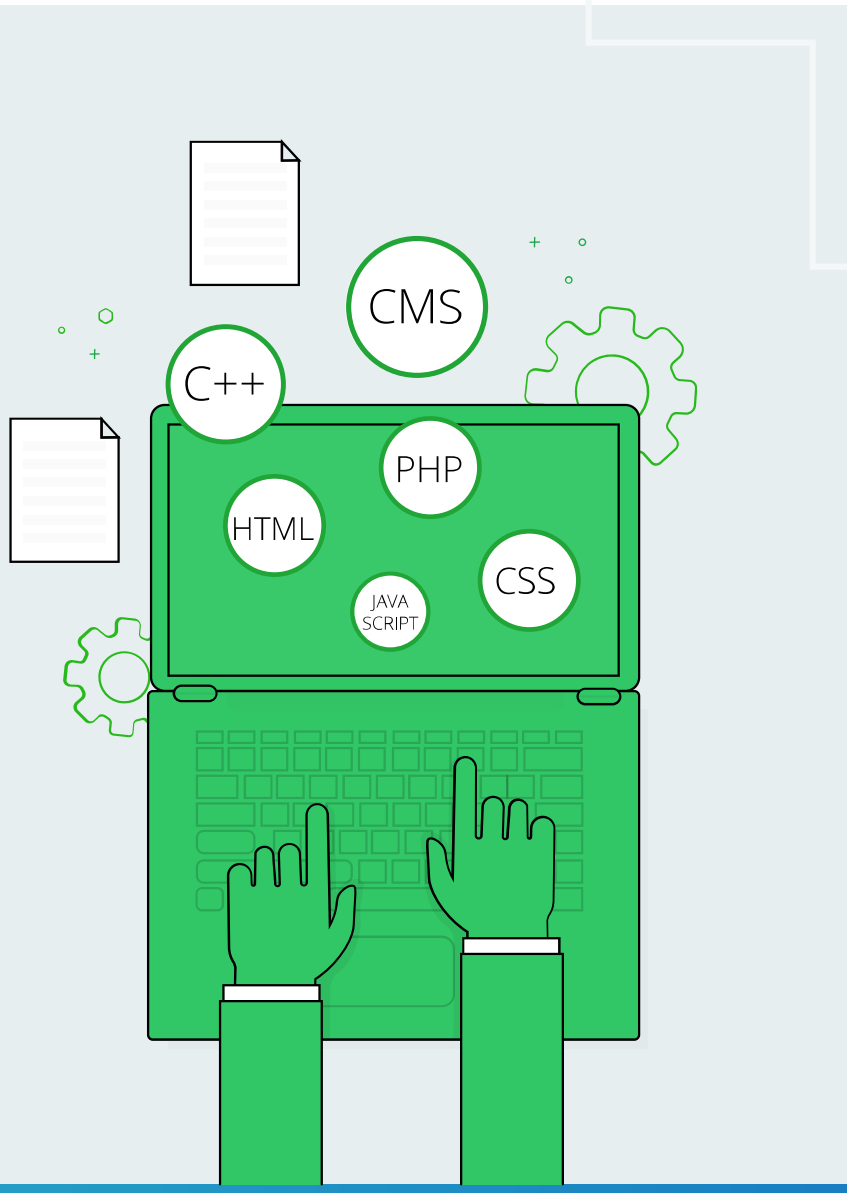
Overview and Best Practices
The most important part of the screening process, skills assessments, are tests conducted to verify the candidate's technical skills and proficiencies.

Timing of the skill assessment may vary, but assessments early in the process help recruiters to advance only the most qualified candidates. For larger organizations with a large volume of applicants, providing assessments early may also cut down the number of less-engaged applicants.

The assessment's ability to evaluate skills depends on the testing environment, reliability of the platform, and the quality of the questions. The most effective approach is to use a remote-first screening assessment solution.

These tools create assessments from a library of questions, pre-built tests, or custom content creation. While building a test, questions should be sortable based on discrete role and skill requirements. Look for tools with randomization features to ensure candidates don't get the same version of the test.

The system should then score the tests, giving recruiters and hiring managers an objective analysis of candidates' skills and helping them screen in and advance individuals who meet the threshold score.



Certified Assessments

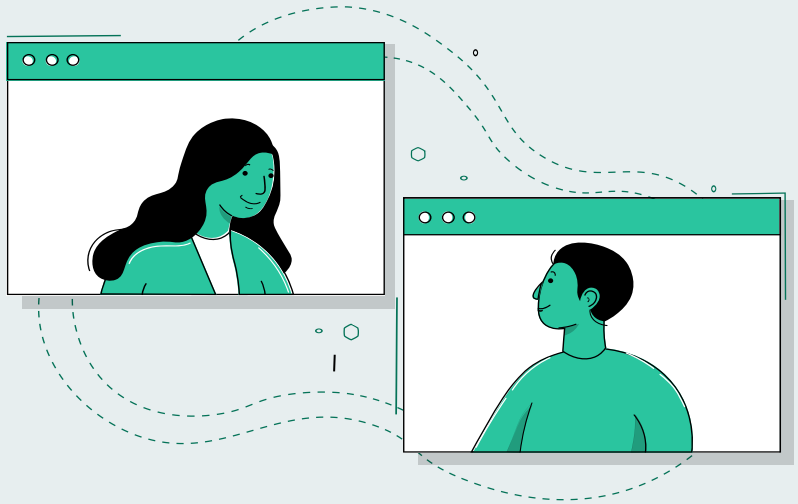
Certification Process and Criteria
Increasingly, many organizations turn to skill assessment platforms with pre-built, ready-to-use testing content that has been certified by experts. Certified assessment content is often produced and reviewed by IO psychologists to ensure that skill assessments evaluate relevant job skills in a consistent and fair way.

Certified assessments provide hiring teams with a streamlined solution which gauges a candidate's technical skills in a manner that meets certain areas of compliance. Criteria include:

Valid - Is the test job-related? Questions accurately assess the knowledge areas and skills that they intend to measure. Questions are based on job-related, critical skills.

Reliable - Is the test consistent? Questions follow a standardized scoring methodology and are tested to measure margin of error.

Fair - Is the test unbiased? Assessments are objectively administered and scored. Fairness and sensitivity reviews remove unconscious bias.



6 **Interview:** Assess Real-World Skills

Technical interviews are fundamental to finding and assessing great software engineering talent. The number and types of interviews conducted will vary depending on role, seniority, and even individual candidates. And the process can be comprehensive.

Google typically conducts three to four rigorous technical interviews in one day. Adobe requires four technical interviews and one HR interview over 6 to 8 hours. The processes at Meta and Amazon are similar, and most organizations, from startups to growth-stage companies, are following suit.

In this section, we'll break down each of the potential interview rounds your company might use during the candidate search.

Coding Interview Round
After a candidate has moved on from the screening stage, they'll typically go through a coding interview that gauges their problem-solving skills and proficiency in languages and frameworks required for the role. The questions also measure their ability to navigate a complete application environment, to translate log messages, and interchangeably use the command-line, IDE, and browser to interact with the environment.

This part of the process is highly customizable, and hiring managers can choose from a number of variations on how to evaluate technical skills thoroughly.

Traditionally, hiring managers conducted project interviews in person, with the candidate solving problems on a whiteboard. In remote technical interviews, candidates don't have the option. Virtual whiteboard tools allow interviewers to assess the candidate's design skills and how they conceptualize a problem.

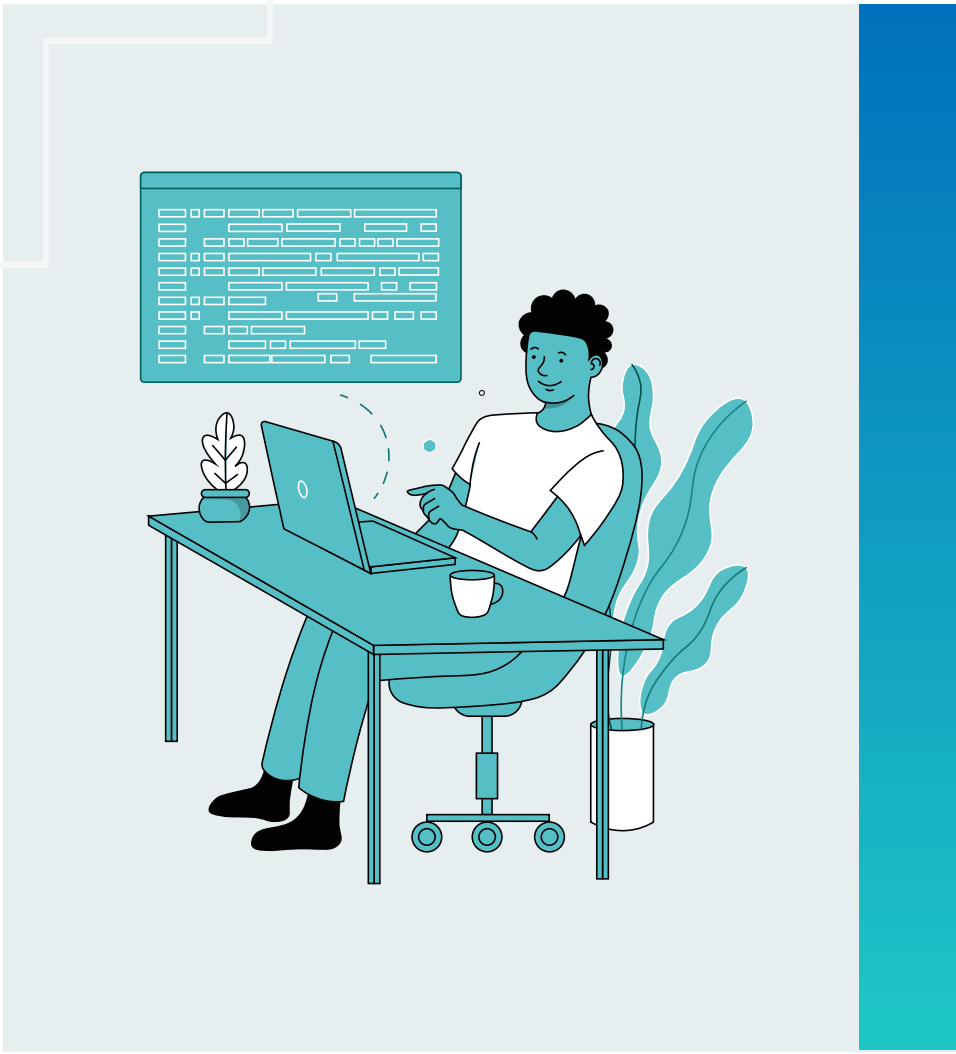
With pair programming tools, hiring teams can watch candidates build on code from pre-screen challenges in real-time, or run and test code together, all in a single session. This enables hiring managers to see a candidate's problem-solving and communication skills in real time — a practical alternative to onsite interviews.

System Design Interviews

In addition to a coding interview, many hiring teams include a system design round. More commonly found in hiring processes for senior-level roles, this stage grants candidates a look into the organization's tech stack and operations, and gives hiring managers an understanding of the candidate's approach to problem solving. These interviews should ideally proceed like discussions, with the candidate thinking aloud about scalability, storage, reliability, and other aspects of the system.

Potential system design prompts include:

- Design a ride-sharing service
- Design a simple social media application
- Design a video streaming service
- Design a URL shortening service
- Design a file-storing and sharing service



Culture and Behavioral Interviews

Culture and behavioral interviews are also common and can occur later in the hiring process. These are carried out with the goal of assessing how well candidates will adapt or add to an organization's culture.

Questions during this round are often open-ended:

- What motivates you to do your best work?
- Have you ever taken a professional risk? What was it?
- How do you respond to criticism?
- How do you prefer to communicate with your colleagues?
- When was the last time you made a mistake at work, and how did you address it?

These questions give insights into the candidates' work style. Seemingly small traits like preferred mode of communication can have a huge effect on their work and collaboration at your company, so it's important to assess their "culture add" and pay ample attention to the soft skills they possess.

You can also round out the cultural fit segment of the interview with open-ended questions that identify a candidate's personal interests, priorities, and appetite for learning and discovery. For example, "Which sites do you visit every day? Which books would you recommend as a must-read?"

DEI Best Practice: Representation Among Interviewers

The interviewers a candidate interacts with are often the only employees they meet during the recruitment process. This is especially true now that most companies have replaced onsite interviews with remote interviews. As such, the main context a candidate will have to assess the diversity of the company's workforce is the diversity of the interviewing panel.

Increasing the diversity of interview panels will make candidates from a range of backgrounds feel represented, while demonstrating a commitment to diversity, equity, and inclusion. It may also have a direct impact on hiring outcomes, as 76% of job seekers and employees consider a diverse workforce an important factor when evaluating companies and job offers, according to data from Glassdoor.

7 **Rank:** Close the Role with the Right Hire

The candidate experience has a major impact on candidate interest in the role and your ability to close. A company that provides a world-class candidate experience will have a competitive advantage against the firms it competes with.

Early Hiring Manager Involvement
One hiring process that some teams follow is to screen, test, and interview candidates before ever having the hiring manager meet with the candidate and sell the role. The risk with that approach is that the candidate might get swayed by half-a-dozen other offers by the time they meet with the manager and learn about the interesting technical opportunies at the company.

The result is a sub-par candidate experience that can lead to recruiters sending offers that few applicants accept.

By involving the hiring manager sooner, (for example, moving the "sell" call with the hiring manager to the beginning of the process), you can foster a stronger candidate experience and ideally increase acceptance rates.

Employer Value Proposition
In the sourcing section, we reviewed the role the employer brand plays in attracting top-tier talent. That work hinges on the development and communication of a clear employer value proposition.

In a traditional, on-site interview, inviting candidates to the office to meet future co-workers in person is a crucial step in the interview process. Candidates can get a sense of the company culture first hand.

When remote interviewing, your interviewees will be able to meet the hiring managers and potential teammates during video interviews — but they won't have an in-person, meet-and-greet experience.

In remote settings, make sure your hiring teams know how to sell the highlights of the role and company. This is particularly impactful when recruiting software engineers, who are among the most scarce and in-demand professionals in the world. Signaling a strong employer brand, value proposition, and culture to candidates is vital to attracting and retaining engineering talent.

Candidate Communication

Touchpoint Planning and Preparing Candidates

Being transparent about the hiring process and what the candidate can expect goes a long way in shaping their experience. Assume every candidate is interviewing for a dozen roles and entertaining multiple offers. What would make that process easier for them, while making your firm stand out?

- Map out how many rounds of interviews the candidate will go through.
- Communicate which skills assessment platform you'll be using.
- Send them a copy of your company's mission and values.
- Share interview prep materials from your skills assessment platform.
- Stay in contact with the candidate from your first phone screen throughout the end of their onboarding process.

Don't be afraid of over-communicating. Lagging on communication is a common mistake companies make which can result in a negative candidate experience — and a lost hire.

Remote and Global Hiring Best Practices

The COVID-19 pandemic, the rise of remote work, and the Great Resignation have had a profound impact on the hiring processes of every organization. While every company has different protocols for in-person and remote interviews, it's important to build your talent acquisition infrastructure as if every interview will be remote. Here are four best practices for creating a first-class remote and global hiring process.

Overcommunicate

While remote interviewing, you won't have the opportunity to create an in-person connection. To ensure that the candidate experience doesn't suffer, over communicate by sending updates at every stage of the evaluation process. This will make them feel valued and ensure a good impression of your employer brand.

Be Transparent from the Start

Make sure your updates are transparent throughout the hiring experience. Remote interviewing can be challenging for many candidates, and you can put them at ease by telling them how the interview process will happen:

- How many rounds of interviews they will face
- What technologies will be used
- What will be evaluated in each conversation or skills test

Eliminate Distractions

During remote interviews, it's important for the interviewer to give the candidate their undivided attention. To eliminate distractions from the interview, establish a quiet environment for the interview, silence notifications, and check that your Internet connection is stable.

Introduce Candidates to the Company Virtually

One of the advantages of in-person interviewing is that you can share first-hand the company's culture and employees. To replicate this virtually, create a branded content package that candidates can review to get a feel for the company's culture and values. This package might include employee profiles of team members or content from recent talent branding campaigns.

Conclusion

If you've made it to the end of this guide, you're now prepared to take on the challenge of recruiting the world's most talented and in-demand software engineers. We've broken down the fundamentals of software engineering, how to find and recruit the best talent, and how to deliver a first-class candidate experience. But there is still work to be done.

The world's appetite for engineers is vast, and the pool of talent to satiate that hunger is finite. And it will remain so for the foreseeable future. For years to come, every new role to fill will be a zero-sum game, a competition between the largest and smallest companies alike. Hiring processes will be dissected, improved, and rebuilt again. Each day will present new challenges, along with endless opportunities.