

The 3 Secrets for Hiring Top Tech Talent (with the Resources You've Got)

How to Go Beyond the Noise
when Hiring Engineers



Table of Contents

Introduction: The paradox of tech hiring today	3
<hr/>	
3 common problems with tech recruiting today—and the secrets to solving them	4
<hr/>	
— Processes that waste your time and resources	4
<hr/>	
— Case study: Slashing engineer time spent on tech recruiting	8
<hr/>	
— Questions that advance the wrong candidates through your funnel	9
<hr/>	
— Platforms that are more frustrating than productive	13
<hr/>	
Conclusion	16
<hr/>	

Introduction:

The paradox of tech hiring today

It's a complicated time to be hiring software engineers. Even with today's uncertain economic climate and high-profile layoffs, companies continue to recruit aggressively for specialized and senior-level technical roles.¹ In fact, job postings for the most in-demand tech roles—software development engineers, backend engineers, and data science managers—more than doubled in 2022 as compared to 2021.²

Yet, technical hiring is harder than it should be. Resource-strapped recruiting teams struggle to source qualified talent for high-priority tech roles, engineers are stretched thin by spending too much time interviewing unqualified candidates³, and candidates themselves become frustrated with what's often a slow and cumbersome recruiting process.

It's time to go beyond the noise by optimizing your tech recruiting to identify the right candidates with the right skills. Doing so empowers you to make the best use of your resource-constrained teams by using technical evaluations that provide an accurate measure of candidates' skills. This guide will walk you through common pitfalls of technical evaluations in 3 key areas—process, questions, and platform—and the secrets to solving them to hire the exceptional tech talent your organization needs to thrive.

¹ [What's actually going on with Google and Facebook hiring freezes? We surveyed 1000 engineers to find out. - interviewing.io](#)

² [Jan–Oct 2022 Dice Tech Job Report - Dice](#)

³ [Four Signs Your Engineers Are Spending Too Much Time Recruiting \(And How To Change That\) - Forbes](#)

3 common problems with tech recruiting today—and the secrets to solving them

Processes that waste your time and resources

Imagine this:

Your company uses multiple rounds of assessments and interviews to fill technical roles, and they aren't easy for anyone. You can tell that your candidates are starting to wonder when it will all end. Your engineers are asking why they have to spend so much time watching candidates code when they'd rather be building software themselves. Your recruiting team is frustrated about having to reschedule tomorrow's interview for the third time today. The IO Psychologists your organization works with are losing sleep over the potential that the whole system is inconsistent and inadvertently unfair to some candidates. And above all, the whole hiring team is worried about how they can possibly get the best candidates through this process before someone else hires them first.

As you can see in the above example, problems with traditional technical hiring processes can occur at all stages of the hiring funnel: from early-stage screening to final round interviews.

Here are some of the most common ways that tech hiring processes can go wrong:



Missing great candidates due to manual resume review.

Many companies use manual resume reviews to screen all applicants at the top of the funnel. Since they need to [sift through a large volume of applications, fast](#), to narrow down their pool of candidates to interview, recruiters must rely on proxies like the prestige of a candidate's previous employers or where they went to school to identify "top" talent quickly. This results in a less diverse candidate pipeline and passing over qualified talent from non-traditional backgrounds.



Wasting engineering hours on early-stage tech interviews.

A common approach for companies recruiting tech talent is to conduct an engineering-led technical phone screen with candidates who pass the initial screening. This approach requires one or more engineers (often the most valuable, senior-level engineers) to conduct an hour-long interview with each candidate who makes it to this stage. It also requires significant prep time for engineers to develop questions and calibrate scoring across interviewers.





Biases and inconsistencies creeping in.

Though not intentional, interviewers can treat candidates inconsistently—especially when pressed for time while trying to juggle their next deadline amidst an interview lineup. Sometimes they don't prepare questions in advance, ask different questions of different candidates for the same role, provide guidance or “[handholding](#)” through challenging technical problems for some candidates (but not others), or change the criteria for evaluating candidates. The result? A biased interview process.



Excessively long recruiting timelines.

A final area that tech recruiting processes can go awry is when they simply take too long. This is painful for both candidates and companies. A drawn-out recruiting process can result in [higher candidate drop-off](#), as qualified candidates may accept offers from other companies that come in sooner. Long recruiting processes also hinder recruiting teams' ability to meet their headcount goals (which often include time to hire), and engineering teams suffer from reduced productivity by having long-unfilled roles on their team.



Secret #1

Optimizing your process for the teams you've got

An optimized and **simpler process** for technical hiring speeds your time to hire, frees up engineers' time, improves the candidate experience, and results in better hires overall for your company.

Consider partnering with a technical interview and assessment vendor that provides more efficient and reliable alternatives to time-consuming recruiting practices. They should offer a top-of-funnel technical screening or assessment to replace manual resume review, freeing up time for a resource-strapped recruiting team. The vendor should also provide a viable alternative to traditional technical phone screens—for instance, by providing this service themselves using research-backed evaluations with unbiased scoring, freeing up time for your company's engineers. Lastly, look for a vendor whose live technical interview platform facilitates [structured interviews](#) to reduce bias in the on-site or panel interview stage.



Case study:

Slashing engineer time spent on tech recruiting

A leading enterprise tech company determined that their engineering teams were spending too much time conducting phone screens and developing custom coding questions, so they pivoted to a new approach: a validated technical screen built and maintained by CodeSignal to replace their traditional technical phone screen.

They have seen the following results by replacing their technical phone screen with a structured and validated tech screen:

\$3 million

Savings in cost of hire for the engineering team

17,800 hrs

Reduction in engineering time spent on recruiting activities annually

45 percent

Improvement in onsite-to-offer rate

* Calculations based on 100 technical hires

Questions that advance the wrong candidates through your funnel

Even when using an efficient process, companies hiring tech talent may waste their engineers' time by using ineffective coding questions in their interviews and assessments.

Consider another scenario:

A senior engineer at your company just spent all last week updating the questions in your coding assessments, and they've already been leaked. The engineering team is in an endless cycle (and time drain) of creating and updating questions, but there never seem to be enough coding questions to keep up with the volume of candidates in your recruiting pipeline. The old questions are too compromised... the new questions may not align with the seniority of the role... and the recruiting team has some questions of their own about the objectivity, consistency, and efficiency of this whole approach.

Sound familiar? This scenario captures some of the ways that the traditional approach to creating and maintaining coding questions can go awry. While engineers are experts at coding and on-the-job scenarios, most have not been trained on assessment design best practices, nor do they have the resources to test the efficacy and fairness of their questions. This can put your company at risk of falling out of compliance with EEOC and other employment regulations.



Let's dig deeper into the most common problems with coding questions in pre-hire technical evaluations:

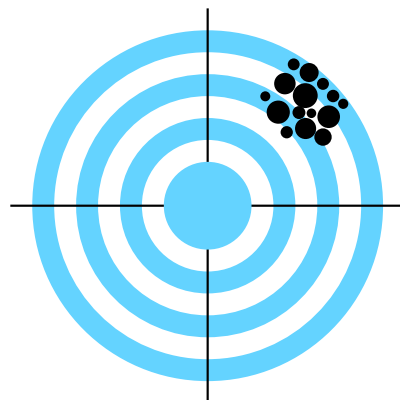


Low validity or reliability.

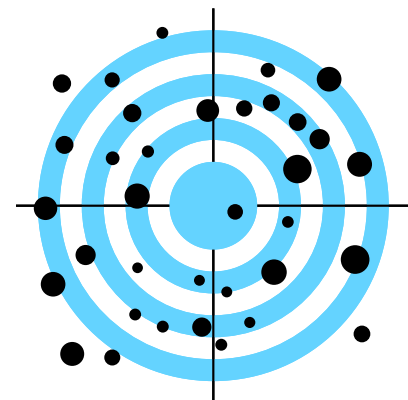
Validity, in the context of hiring assessments, refers to how accurately a coding question or technical assessment measures the specific technical skill that it's intended to measure. Reliability, on the other hand, refers to consistency. Does a coding question consistently measure the skill it's intended to, across a large volume of candidates? If not, that question has low reliability.

How is validity different from reliability?

It can be helpful to think of validity and reliability using the visual of a dartboard. A coding question with low validity produces consistent results across candidates—but the skill it measures isn't the one it's meant to. The darts all hit in the same place each time, but not in the bullseye. A coding question with low reliability identifies candidates with the right skill only some of the time. In the dartboard analogy: the darts land all over the board—sometimes hitting the mark, sometimes not.



Reliable, Not Valid



Unreliable, But Valid



Poor alignment to the seniority of the role.

Technical evaluations often ask generic algorithmic questions because it's easy to grade these questions in any language, and variations are relatively simple to generate. [While algorithmic questions are appropriate for evaluating core programming knowledge](#) among new grad and early career candidates, they are much less relevant to senior-level candidates who are years past their formal CS education. Experienced and senior-level candidates end up frustrated if asked only these types of questions, which may have little relevance to their real jobs.



Unclear or bias-laden question wording.

From the ordering of questions to the phrasing and content itself, writing interview questions is an art. When professionals aren't involved to develop these questions, the writing might suffer from a lack of clarity or use [culturally-specific language and examples](#). This can be alienating and confusing, especially for people from different backgrounds and with different levels of English language ability.



Questions that get leaked as soon as you launch your assessment.

It's extremely common for coding questions to be leaked on platforms like LeetCode and Blind. Once coding questions are leaked, it's challenging for hiring teams to know if the candidate has seen the question somewhere else already. To keep a steady stream of questions and variations (and replace questions when they're leaked), engineers need to maintain different versions of their coding assessments. This requires hundreds of hours of work each year and takes away from time spent on product development.



Secret #2

Creating questions that assess the right skills, accurately

Smarter questions for technical assessments and interviews should be written by subject-matter experts (SMEs) and validated by Industrial-Organizational (IO) Psychologists. Throughout an assessment or interview, questions should build in complexity and not require too much context switching for the candidate.

Engaging a vendor whose team includes both technical SMEs and IO Psychologists is the easiest way to ensure that your coding questions will provide a strong signal of candidates' skills, align to the roles and seniority levels you're hiring for, and avoid biased wording. At best, the vendor should build validated technical assessments that include [many variations of equivalent difficulty](#) to mitigate the impact of question leaks and reduce plagiarism in your technical evaluation process.

Platforms that are more frustrating than productive

Finally, tech recruiting can go wrong when a company uses a platform or tools that work against them. This can happen when a platform fails to capture the full breadth and depth of candidates' job-relevant skills—or that isn't designed for technical hiring at all.

Let's consider one more all-too-familiar story:

You've got five technical hiring tools in place at your company, across three different teams. None of them has all the features you're looking for. And none makes hiring simple for your resource-constrained teams. Despite the hiring team's best efforts, their interviews and assessments still don't match how software engineers actually work. So it's not a great experience for candidates. And it's not giving your engineers a very accurate picture of how those candidates would actually perform on the job.



A poorly-designed tech hiring platform, like the tools described in the scenario above, have a few key tells. Here's what to look out for:



Hard-to-use IDE.

A platform that's glitchy, or an integrated development environment (IDE) that's uncomfortable, unintuitive, or lacking the expected features, means that candidates will need to spend time trying to figure out how to take your technical evaluation rather than focusing on the questions. This can result in candidates feeling frustrated—or worse, like they weren't given a fair shot to demonstrate their skills. For companies, this may mean a higher candidate drop-off rate, with qualified candidates opting not to continue in your process. Another potential outcome: false negatives. Qualified candidates may perform poorly on your assessment due only to the difficulty of using a bad platform.



Inability to assess job-specific skills.

Many technical hiring platforms provide a basic coding environment, but lack the full functionality that engineers and developers need to build real software. Many cloud-based IDEs do not provide candidates with essential features like terminal access, fully functional autocomplete, or debugging tools. They may also lack tools that are specific to specialized technical roles: live frontend preview for frontend developers, or Jupyter Notebook for machine learning engineers, to name a few. This means that hiring teams can only assess a fraction of a candidate's job-relevant skills.



Lack of key tech stack integrations.

Recruiting teams rely on a variety of tools to manage their candidate pipelines: scheduling automation tools, sourcing tools, and—at the center of it all—an applicant tracking system (ATS). When a technical assessment and interview platform doesn't integrate with your company's ATS or other key recruiting tools, the result can be inefficiencies across your entire tech recruiting process—particularly if your recruiting team has been downsized.



Secret #3

Finding a platform that works for you and your candidates

A stronger platform for technical hiring should include an [advanced, intuitive IDE](#) that simulates real-world software engineering work and should integrate seamlessly with your existing tech stack. A best-in-class platform will provide candidates all the tools and functionality they need to code like they would on their local machine. It will also allow your hiring team to accurately evaluate all the technical roles you're hiring for—with solutions for top-of-funnel skills assessment, full-service technical screening, and live on-site interviews.

Conclusion

In spite of persistent high demand for technical talent, traditional tech hiring methods are noisy because they fail to provide hiring teams with a strong signal of candidates' skills. In this guide, we've identified three key areas tech recruiting goes wrong: processes that waste your time and resources, questions that advance the wrong candidates through your funnel, and platforms that are more frustrating than productive.

There are three secrets to go beyond the noise in technical recruiting:

- A **simpler process** that speeds your time to hire, frees up engineers' time, improves the candidate experience, and results in better hires overall
- **Smarter questions** that are written by subject matter experts and validated by IO Psychologists
- A **stronger platform** that supports you (and candidates) at every step in the hiring process, plugs seamlessly into your ATS, and is supported by an industry-leading IDE that simulates real-world software development

With these in place, your company is well-equipped to make the exceptional technical hires that will take your engineering team to new heights.



Fortunately, you don't need to take this on alone.

As the leading technical interview and assessment solution, CodeSignal helps companies go beyond the noise with smarter assessment questions, a simpler process, and a stronger platform.

CodeSignal's [Pre-Screen](#), [Tech Screen](#), and [Interview](#) provide advanced job simulation technology across the entire hiring process with a framework-based approach so teams can conduct fair and predictive evaluations, save valuable engineering resources, provide a better candidate experience, and hire the right talent, fast.

To learn more about how CodeSignal can help you optimize your technical hiring processes and land top talent, schedule a discovery call today.

[LET'S TALK](#)

