Optimizing Your
Tech Hiring Practices:

How to Drastically Reduce Engineering Time Spent on Recruiting

CodeSignal

Introduction

Technical recruiting presents a unique challenge: in-house engineers often play a major role in hiring, above and beyond that played by the recruiting team. One estimate, in fact, finds that the engineering team incurs **6 times the cost** that recruiting does for each technical hire due to engineers' high salaries and the number of hours they spend on recruiting activities.[1] Engineering managers spend 15 percent of their time just on recruiting.[2]

When engineers spend a considerable portion of their time on non-engineering tasks, there is a significant impact on their **productivity and innovation**. When engineering teams are burdened with constant interruptions during their work day to interview and assess candidates, they are less able to engage in the deep work that is essential to their work: solving complex problems, developing innovative solutions, and ideating new products.[3]

The cost of engineering time spent on recruiting has yet to make its way into most recruiters' calculation of cost per hire and other core metrics. **It's time for that to change.**

[1] You probably don't factor in engineering time when calculating cost per hire. Here's why you really should. - interviewing.io blog

[2] A Day in the Life of an Engineering Manager | Toptal

[3] Think deep work isn't relevant to software engineers? Think again - Uplevel on the productivity impact of distraction culture

This guide will help you identify and measure how much time your engineers spend on technical recruiting, as well as how to drastically cut down these hours to optimize for key recruiting metrics. We break this down into 4 common recruiting resource drains on engineers' time:

01. Technical interviewing

02. Creating coding questions

03. Managing leaked questions

04. Evaluating candidates

By the end of the guide, you'll have gained practical strategies you can implement today to reduce engineering hours spent on recruiting—without sacrificing quality of hire.

Let's get started.

Common recruiting resource drains on engineers—and how to stop them

It's inevitable that engineers will spend some of their time on recruiting activities. After all, you wouldn't want to bring on new software developers or engineers who your current team hasn't even met! However, too much time spent on recruiting can be a drain on engineers' productivity.

To help your engineering team optimize their productivity and capacity to innovate, recruiting leaders can develop technical hiring processes that minimize the time and energy engineers spend on four common resource drains: conducting technical interviews, creating coding questions, managing the impact of leaked questions, and evaluating candidates.

Technical interviewing

The first and most obvious resource drain on engineers' time is **time spent conducting interviews**. Depending on your organization's recruiting process, engineers may be involved in technical phone screens, onsite panel interviews, executive interviews, secondary technical screens, or all of the above.

Traditional technical phone screens, for instance, require senior-level engineers to conduct an hour-long interview with each candidate who passes the initial screening. Given industry average salaries for senior engineers, we can assume the following cost per technical phone screen:

(1 hr interview + 0.5 hr interview prep + 0.5 hr interview debrief) x $200/hr salary = $400 per phone screen.

If you expect to conduct 40 technical phone screens to fill one position, that's a cost of $16,000 per hire just for this recruiting stage.

Solution:
Structured technical screens

Rather than rely on in-house engineers to conduct early-stage interviews like technical phone screens, companies are saving engineering hours by engaging a validated, asynchronous technical screening solution.

Here's a few things to look for in a technical screening service or vendor:

Assessment validation by IO Psychologists

Automated scoring to ensure objective, unbiased candidate evaluation

ATS integration to save time for recruiters and hiring managers

Best-in-class technical screening services like CodeSignal Pre-Screen provide an objective, accurate signal of candidates' technical skills while saving engineering teams hundreds of hours they would have spent conducting interviews.

Case study:
Using structured technical screens at scale

A leading enterprise tech company determined that their engineering teams were spending too much time conducting phone screens and developing custom coding tasks, so they pivoted to a new approach: a structured 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

Creating coding questions

Apart from the time they spend conducting live interviews, engineers may also spend many hours developing coding questions for technical assessments, take-home assignments, and live coding interviews.

Developing a single coding question can take an engineering team anywhere from 2 to 6 hours, depending on the complexity of the question and accounting for the time additional engineers spend reviewing the question.

When we know how many hours engineers must spend creating new coding questions, we can calculate the cost to engineering of creating an assessment or interview template:

4 average hours per coding question x 4 coding questions (recommended average) x $200/hr salary = $3,200 cost for coding question creation for just one role

Another challenge: While engineers may feel confident in writing a coding question, they may overlook other key steps needed to write a question that will accurately, reliably, and fairly predict which candidates have the skills to succeed on the job. Engineers often underestimate the difficulty of the question they are asking, for instance, or may choose "favorite" questions over ones proven to provide an accurate signal of a candidate's job-relevant skills.

The best coding questions are developed via a collaboration between subject matter experts (SMEs) and Industrial-Organizational (IO) Psychologists trained in best practices for assessment design—not by engineers alone.

5 steps for writing a coding question
(that engineers might overlook)

01. **Conducting a job analysis** to identify the knowledge, skills, abilities, and other characteristics (KSAOs) required for the job, then creating a coding question that evaluates these KSAOs

02. **Validating** the coding question with IO Psychologists and SMEs, who together can assess if the question is predictive of job-relevant skills

03. **Piloting** the question with developers or engineers to ensure they interpret the question as intended

04. **Conducting adverse impact analysis** to verify that the question is unbiased and fairly assesses candidates across protected demographic categories.

05. **Ongoing monitoring** to guarantee the question continues to serve as a valid measure of candidates' skills and adjusting if needed

Solution:
Validated pre-built questions

It is possible to develop great coding questions that accurately and reliably predict candidates' skills, *without* requiring the engineering team to spend hundreds of hours a year creating and validating new questions.

To do this, we recommend partnering with a technical interview and assessment vendor that offers **expertly-written skills evaluations** that are developed and validated by IO Psychologists and SMEs. This can reduce the time your engineers spend creating coding questions to zero.

Managing leaked questions

For companies that create and maintain their own coding questions, managing leaked questions can feel like a never-ending game of "whack a mole." As soon as a leaked question gets taken down from one site, it shows up on a new one. One source finds that even when candidates sign non-disclosure agreements (NDAs) at the start of the hiring process, interview questions still get leaked.

Research with CodeSignal customers found that, before implementing CodeSignal's framework-based assessments, cheating and plagiarism hit an all-time high 3 months after a company starts using a coding assessment. This means that companies who create their own coding questions can expect that their engineers will **need to rewrite them 4 times a year, which requires approximately 400 hours of engineering work.** This can be incredibly frustrating for the engineers tasked with creating and maintaining coding questions for technical assessments and interviews—and costly for your business.

400 hrs rewriting leaked questions x $200/hr salary = $80,000 annual cost of coding question maintenance

Solution:
Skills Evaluation Frameworks

Managing leaked questions is a tricky endeavor. Closely monitoring sites like LeetCode for your coding questions and requesting DMCA takedowns is simply not scalable or sustainable for most companies.

The solution is not to eliminate question leaks, but instead to mitigate their impact. At CodeSignal, we do this through developing Skills Evaluation Frameworks that use **dynamic question rotation** to prevent any two candidates from seeing the same coding questions in their assessment or interview. Dynamic question rotation requires you to have many variations of each coding question that are consistent with each other in terms of difficulty, face validity, and pass rate.

For a company's in-house engineers, this would require hundreds of additional hours of work creating coding questions. Instead, we recommend partnering with a vendor that offers technical interviews and assessments with a dynamic rotation of pre-built questions. The IO Psychologists and Assessment Design Engineers who make up our Talent Science team develop **hundreds of minor and major question variations** with careful testing to ensure fair and consistent scoring, resulting in over a million variations of CodeSignal's frameworks.

Evaluating candidates

The final recruiting resource drain on your engineering team is the time they spend manually reviewing candidates' code and assessing their fit for the role. Depending on your company's recruiting process, your engineers may spend hours evaluating each technical candidate:

Technical phone screen debrief: 30 minutes

Take-home assignment scoring: 30 minutes

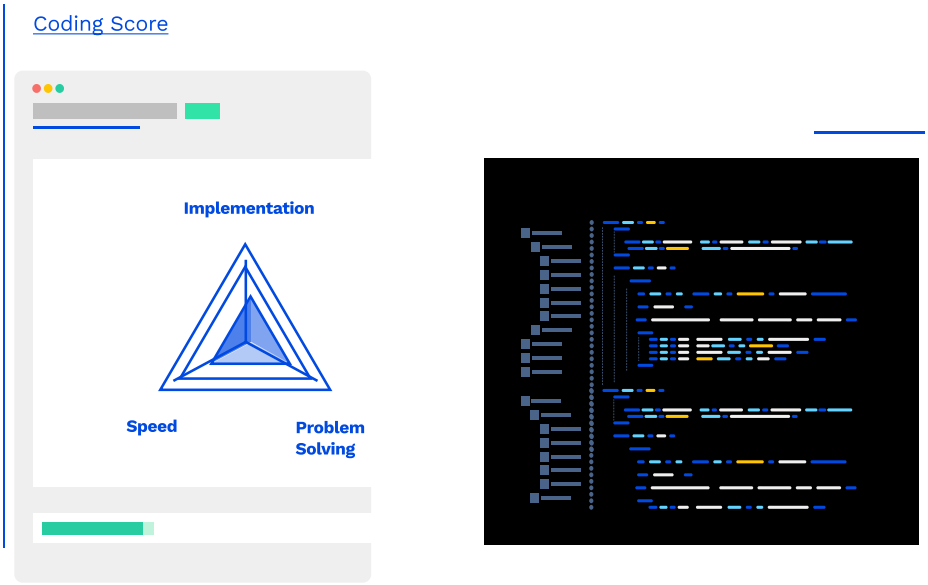
Onsite interview debrief and calibration: 30 minutes

The engineering cost of manually evaluating candidates adds up quickly. Let's take the technical phone screen debrief, for example. Assume that to fill one role, you'll conduct 40 tech screens—that's 20 engineering hours spent just on the tech screen debrief.

Solution:
Objective coding scores and structured rubrics

Companies can drastically reduce engineering time spent evaluating candidates by 1) using a research-backed tool that **scores candidates' code automatically** and 2) implementing **structured rubrics** into their debrief process.

Many technical hiring platforms available on the market automatically score candidates' code. However, these often provide limited information about a candidate's skill. Companies who evaluate their technical candidates using CodeSignal Pre-Screen receive a **comprehensive coding report** for each candidate. This report includes a validated Coding Score that accounts for 4 factors in its calculation: correctness of the solutions, implementation ability, problem-solving ability, and speed. CodeSignal's Coding Score allows companies to quickly and objectively screen a high volume of candidates, without requiring engineers to spend time manually reviewing code.



This granularity of the Coding Report allows companies to ensure candidates that advance to final-round onsite interviews are highly qualified. With fewer unqualified candidates making it to this stage, onsite interviews are more productive and allow engineers to dive deeper into a candidate's problem-solving and troubleshooting abilities.

To ensure consistency, limit bias, and get a full picture of each candidate's skills, we recommend using a **structured rubric**, a scoring guide used to evaluate a candidate's knowledge and skills, to debrief each candidate after an onsite or panel interview.

Rubrics not only allow hiring teams to make fairer and more objective decisions—they also save engineers' time by keeping the debrief conversation focused and providing a reusable template for each candidate onsite evaluation.

4 principles for developing a rubric
for an onsite interview

01. **Use a numerical system.** Numbers allow you to calculate a final score. Without numbers, you risk ending up in the nebulous territory of “pretty good” and “okay.”

02. **Spell out what each score means.** If you’re using a range of 1-4 for each item in your rubric, for instance, give a clear description of what counts as a “3” coding syntax score versus a “4.

03. **Include both technical and communication skills.** Collaboration and communication may be “softer” skills, but that doesn’t mean they shouldn’t be scored on the rubric. If they matter for your decision-making, include them on the rubric.

04. **Calibrate your rubric.** We recommend having all interviewers on your team score the same interview independently. Then they’ll compare scores, discuss differences, and repeat (if time allows). This process will help ensure that everyone is using the rubric in the same way.

Conclusion

This guide has taken you through 4 common ways that engineers spend time on recruiting activities, as well as changes you can make to your technical hiring process to address these. Doing so can save your engineering team hundreds of hours—which can amount to over $3M dollars in savings, as we saw in one example above. To recap, these 4 areas for change are:

01.

Technical Interviewing

Instead of having engineers conduct multiple one-on-one and panel interviews with technical candidates, replace early-stage technical phone screens with a validated, asynchronous technical screening solution.

02.

Creating coding questions

Engineers may enjoy writing their own coding questions—but the time they spend doing so adds up quickly, and they often overlook key steps needed to validate and pilot their questions. Engage a vendor that offers rigorously researched, pre-built assessments instead.

03.

Managing leaked questions

Spare your engineering team the tedious work of rewriting questions that got leaked online. Instead, use assessments and interviews powered by skills evaluation frameworks that use dynamic question rotation to ensure no two candidates see the same coding questions.

04.

Evaluating candidates

With the technical assessment technology available today, there is no need for engineers to manually review candidates' code. Instead, choose an assessment and interview tool that offers a validated coding score, and introduce structured rubrics into the debrief process to reduce engineering time spent evaluating candidates.

Reducing engineering time spent on recruiting has tremendous cost benefits for your company. An additional and often overlooked benefit is the **impact of having more time on engineers' productivity and innovation**. When engineering teams are freed from the burden of recruiting tasks that frequently interrupt their work day, they unlock a much higher capacity to engage in the deep work that ultimately drives innovation and delivers value for your company.

Get started

Ready to significantly reduce cost
per hire and free up your engineers
to do the work they love?
The technical hiring experts
at CodeSignal can help.

LET'S TALK