karat^

How to Build Better Technical Interview Questions

How to Build Better Technical Interview Questions

A good technical interview teaches us something about the candidate, and it should be exclusively things that relate to how well that person is likely to perform in the job they've applied for. Interviews need to be consistent and scientific to achieve this, and the way we craft interview questions plays a significant part.

To make interviews more consistent and scientific, we should try to firmly separate the things which can be objectively measured from the things that cannot. By building technical interview questions to focus on objectively measurable skills, we can have a clearer idea of how candidates accurately compare to each other. For subjective considerations, we can make cross-candidate comparisons more viable by focusing on consistency in the interview questions we ask, as well as how we evaluate candidate responses.

Ultimately, successful technical interviews are predictive of the candidate's job success, fair to candidates and to our organization, and even enjoyable for candidates. In this guide, we'll show you how to create successful technical interviews by applying a scientific approach and consistency to interview questions.

karat^

Designing Interview Questions to be Predictive

Figuring out if a candidate is right for the job is obviously what a job interview is all about for most of us. But let's be thoughtful about how we define *right for the job*: very specifically, this should mean being *able to be successful in the role.*

But what does success look like?

In theory, we *should* be able to turn to the role's job description, which *should* clearly and unambiguously describe what "success" means. In reality, it's rare for job descriptions to be that revealing. So where can we turn?

In the world of assessment science, we'd begin by performing a job task analysis, or JTA. As the name implies, that study would reveal the key job tasks needed on the job. A good JTA will look at a cross-section of criticality and commonality. In other words, it'll show us which tasks are most commonly performed on the job — the ones that incumbents in the role perform day in and day out — as well as the ones that are most critical to the role. Those latter tasks are the ones that, if performed wrong, will cause the highest level of negative impact on the organization. With a good JTA in hand, you can select a group of tasks that are both common and critical and focus on those in the job interview. That intersection between commonality and criticality shows what "success" truly looks like in the role.

But here's the trick: without a JTA, or something substantially like it, you may have no idea what success actually looks like. The sad reality is that most job interviews start from a "gut feeling" of what's required to be successful in a role. Even more sadly, many job interviews are built not around what "success" looks like, but instead around topics that weren't well understood by people that the organization perceived as being unsuccessful in the role.

Many times, organizations build their job interviews **without a strong understanding of what success looks like**. Instead, they consider past incumbents who they feel were unsuccessful, and build an interview that those past employees wouldn't be able to pass.

Constructing a decent JTA doesn't need to be hard, and it's the pivot point where an interview turns from exclusionary to inclusionary.

1. Start by polling your incumbents — all of them, but especially those who routinely meet expectations in their performance reviews. Ask them to create bullet lists of the tasks they perform. Don't worry about commonality or criticality at this stage, but *do* worry about specificity. For example, software engineers might be tempted to just put "write code," but ask them to dig a little deeper. What kind of code? What does the code actually do? Do they write code that maintains state, that scales in a concurrent environment, that accesses databases, that secures data? Sometimes, putting a group into a conference room with some pizza for a couple of hours can be a useful way to come up with this list.

   Notably, "soft skill" tasks can and should be included. That might include things like "presenting at stand-up meetings," "constructing status slide decks," and so on.

2. Continue by asking one or two team members to de-duplicate the combined list. In many cases, you'll find you have the same tasks listed multiple times, just worded a bit differently, and it's a quick exercise to consolidate them.

3. Send the consolidated list around to everyone who currently sits in the role. For each task, ask them to rate the commonality of each task on a scale of 1 (I almost never do this) to 5 (I do this multiple times a day). Also ask them and their managers to rate the criticality of these tasks, again on a scale of 1 (it's not a huge deal if this is done wrong, perhaps because it's so easy to catch or fix) to 5 (the world will explode if this is done wrong).

4. For each task, average out the commonality and criticality responses.

5. For each task, multiply criticality by commonality. You'll now have rankings from 1 to 25 — sort these in descending order.

6. Your top 10 tasks are the ones to consider for an interview. Some of these may not be feasible to actually evaluate in an interview, and that's fine — just keep moving down the list until you have 10. In reality, 10 will be too many to properly evaluate, but it gives you a little flexibility in constructing the final interview.

With your JTA in hand, you now have a much more scientific and data-driven idea of what "success" looks like in a role. If a candidate can successfully demonstrate their ability to perform those tasks in an interview, then there's a good chance they'll be able to perform those tasks well on the job too. Perhaps more importantly, you've taken an important first step in removing bias from the interview, because you're focused solely on job-related tasks.

Now let's take a look at one of the first principles of interviewing: each thing we examine in an interview, meaning each question we ask or task we consider, should teach us something new about a candidate. For example, let's suppose we have three questions, identified as A, B, and C. Over time, we might notice that candidates who always answer A correctly also answer C correctly, and that candidates who get A wrong always get C wrong. In that case, A is said to correlate strongly to C, and we would modify the interview so that we asked one or the other, but not both. After all, if a candidate always gets them both or misses them both, then asking both doesn't teach us something new, right? We only need to ask one of them to learn what we're going to learn — and we free up time to ask a different question that does teach us something new.

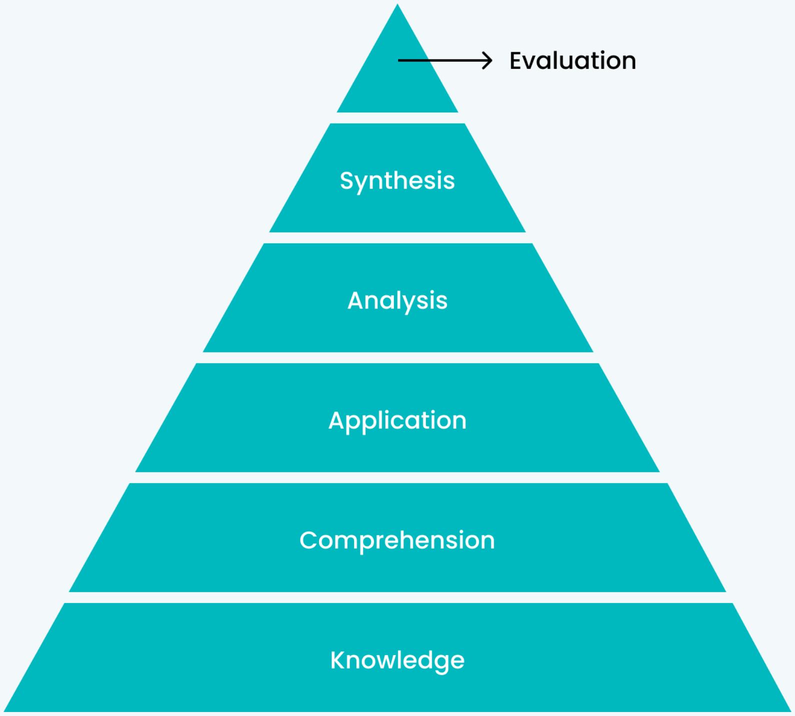
That correlation principle can actually be measured over time as you interview candidates, which is why it's important to keep track of interview results. Correlation is measured on a scale of 0 to 1. In our above example, A would be described as having a correlation of 1 with C. Now consider a case where half the candidates who get A right also get B right, and vice-versa; A would be said to have a .5 correlation with B. Typically, you would drop any questions that have a correlation of .8 or more with a question you're already asking. That is, if 80% or more of candidates always get or miss them both, then don't ask both.

Correlation theory is a great example of how interviews should evolve and improve over time. Out of the gate with a brand new interview, it's tough to guess where correlation might occur. It's not until you have data that you can start improving. And of course, your interviews need to be consistent, so that the data you're collecting can be used to make these kinds of improvements.

In the end and over time, your JTA-based interview will do a better and better job of identifying high-potential candidates. Of course, you do need to hold yourself accountable by tracking the performance of candidates who pass your interview and are hired. At their next performance review, does everyone agree that they're doing well? If so, then you can be pretty confident that your interview was, in fact, predictive of success.

Bloom's Taxonomy of Cognitive Activity

Doctor Benjamin Bloom developed Bloom's Taxonomy of Cognitive Activity to describe differing levels of thinking. These levels are generally depicted as a pyramid, with the lower-order activities on the bottom and the higher-order activities on the top:



Importantly, each level includes the ones below it. That is, in order to apply something, or put it to use, you must also comprehend (or understand) it, which means you must also know about it.

In the context of a job interview, you'll generally want to aim for Application-level questions and Analysis-level questions. Interestingly, Analysis-level questions (which, remember, include the ability to apply something) often take less time and are more predictive of future outcomes. For example, in a software engineering interview, Application might look like asking a candidate to write code. But Analysis might involve looking at an existing code example and commenting upon it, perhaps to suggest improvements. In that example, reading code actually takes less time than sitting down and laboriously typing code — and engages more of the candidate's cognitive powers, and is more predictive of eventual job success.

A number of institutions publish online lists of "Bloom's Verbs." These are designed to help you quickly figure out which level of Bloom's a given interview question targets. For example:

💡	List the six different types of software test.	*This targets Knowledge, requiring a candidate to do nothing more than regurgitate a fact they memorized.*
💡	Explain the purpose of a unit test.	*This targets Comprehension, requiring a candidate to understand a fact, but not necessarily how to use it.*
💡	Write a unit test.	*This targets Application.*
💡	For a given situation, select the best type of software test to use.	*This targets Analyze, testing whether a candidate knows when and how to use something.*

It can be a little unintuitive to construct interview questions at the Analyze level, because as humans we tend to have a gut instinct that seeing someone do something is the best possible test of ability. But in fact, job success often depends more on knowing when and why to do something than specifically knowing how to do that thing — given that the "how" is often easily learned.

Designing Interview Questions to be Fair

We want interviews to be fair, right? Sure, we do. But what do we actually do to ensure the fairness of an interview?

We should probably start by defining *fair.* One dictionary definition is:

> *adjective,*fair·er, fair·est.
>
> 1. free from bias, dishonesty, or injustice:
> 2. *a fair decision;*
> 3. *a fair judge.*

Free from bias is probably a good, concise definition of the term for our purposes. So how do we ensure that our interviews are free from bias?

Structure. Consistency. A laser-like focus on job tasks, to the exclusion of as much else as possible. A relentless focus on what actually matters on the job, and a willingness to rethink what truly matters for job success.

You're probably already aware of the concept of unconscious bias, and the role it can play in processes like job interviews. Having unconscious biases doesn't make someone a bad person, it simply makes them a person. A human being. We're all hardwired to feel more comfortable around people like ourselves — people who look similar, speak similarly, and who exhibit the same values that we prefer in ourselves. Having these unconscious biases doesn't mean we harbor any ill will toward people who look, speak, or think differently — it simply means we're human. That said, we can acknowledge those unconscious biases and work to keep them from coming into play during what should be a fair, equitable, and bias-free interview.

Structure: Having an interview that is structured is a good starting point. Structure means that our interviews have a fixed format. We might spend 10 minutes discussing one topic, and then move on to another topic. We might focus on questions that target the Analysis level of Bloom's taxonomy. We might start the interview with 5 minutes of introductions to help a candidate feel more at ease. Whatever our structure is, the point is to have one. Ideally, we even share that basic structure with candidates up front, because doing so will help them feel more confident, and enable them to better bring their best selves to the interview.

Consistency: A critical element of fairness is treating every candidate the same, and that means conducting every interview the same. Now, this doesn't necessarily mean we have only one set of interview questions — we do need to be aware of the fact that, especially in large, well-known organizations, the questions we ask during interviews will wind up on the Internet somewhere. It's okay to have multiple different "interview sets," so long as they are equivalent. Meaning, each interview set targets the same job competencies, targets the same level of Bloom's taxonomy, and follows the same structure. Consistency is how we ensure that the scores we give to one candidate are directly comparable to the scores given to another candidate, allowing "apples to apples" comparisons.

A focus on job tasks: While it can seem more personable to ask a candidate about their hobbies, or their families, or the neighborhood where they live, those topics have no relation to a candidate's ability to perform a job, and therefore shouldn't be included in a job interview. Once you hire someone you can start to get to know them as a person; during the interview, focus solely on their ability to perform the most common and critical job tasks — ideally, tasks you've identified in a valid job task analysis.

Objectivity: To the maximum degree possible, score candidates using a rubric, or scorecard, that provides highly detailed guidance and absolutely minimizes interpretation by individual interviewers. This is truly the key to removing bias. And here's a good exercise: After you've come up with interview questions, have two or three existing team members answer those questions in writing. Then, share those answers — without the team members' names attached — with several prospective interviewers, and ask those interviewers to use your proposed rubric to score those answers. If you don't get the same scores from each interviewer — if you have poor inter-rater reliability or p-value — then your interviews will almost inevitably contain at least traces of unconscious bias. Consider reworking your rubrics and trying again, until you can get a better p-value across interviewers.

Also consider the context of your interview questions. For example, if you work at a large bank, it can be tempting to frame interview questions in the context of banking. After all, doing so can feel very “real world,” right? But when you do that, you’re automatically adding a bias toward candidates who have already worked in that industry. If that bias is intentional — for example, perhaps your job description actually specifies a requirement for prior banking experience — then it’s fine to acknowledge it and continue. But if your job description doesn’t specify a requirement for that prior experience, then you’re creating a strong bias against candidates who had no idea to expect that kind of specificity in their interview.

You can perform a kind of “bias check” on your interview questions and scoring rubrics: sit down as a team and ask yourselves, “is there anything in here that might require some background, experience, or other information that is unique to us, and not specifically demanded in the role’s job description?”

Designing Interview Questions to be Enjoyable

Can an interview be enjoyable for a candidate?

Rather than blithely answering “yes” or “no,” consider this:
They’d better be.

A job interview is your first encounter with what will hopefully be a future colleague. A coworker, and possibly even a friend. Certainly a member of your professional network. With all of that on the line, don’t you want to make a good first impression? You do, as the saying goes, only get one chance to make that first impression, and you’re going to make it long before that person’s first day on the job.

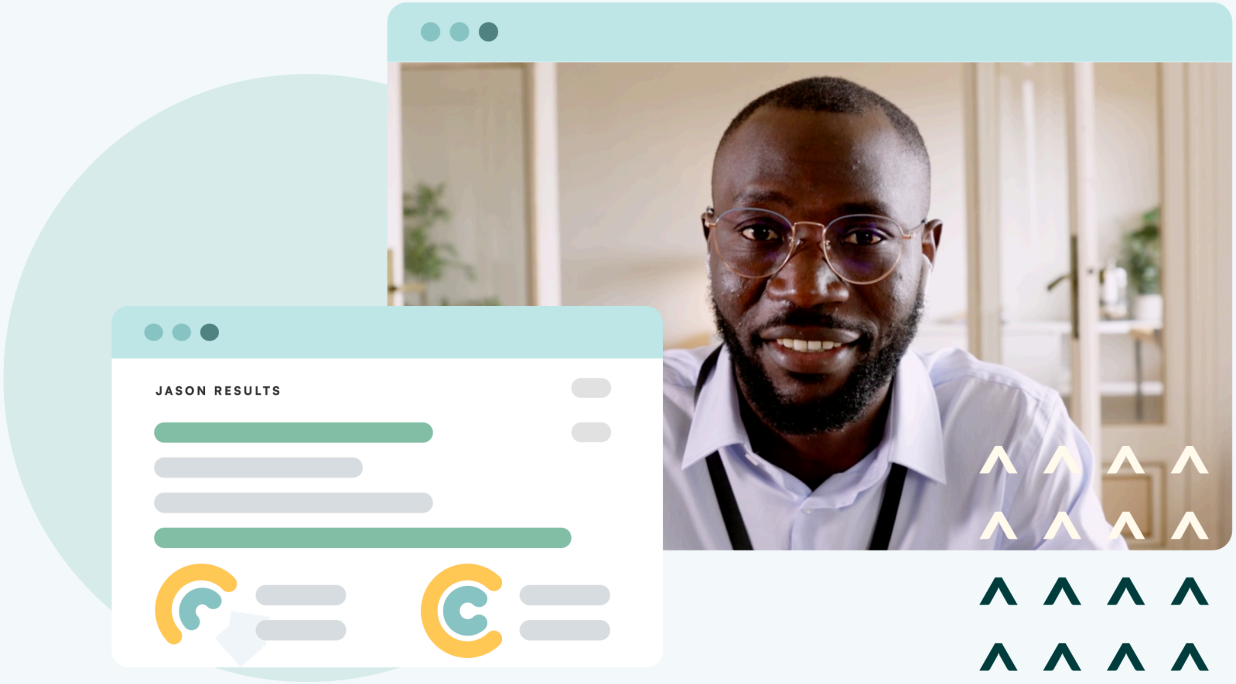
A job interview also says a lot about your organization, what it values, and how it treats people — and please believe that candidates talk. Numerous websites — beyond the obvious ones like Indeed and Glassdoor — offer candidates a chance to anonymously share their impressions about organizations’ interview approaches, and candidates are rarely shy about clearly stating their feelings. Organizations with a bad interviewing reputation tarnish their employer brand, and they make it harder for themselves to recruit qualified talent.

So your interviews had better be enjoyable.

Many of the tactics for creating an enjoyable interview, therefore, involve minimizing the stress of interviews while firmly realizing that we can never, ever fully eliminate it and see candidates performing at their peak.

One way to do this is to connect to the job. With every interview question, try to provide a little bit of context about why that question matters to you. For example, "The role you're interviewing for works on both back-end code and middleware code. We run in an extremely high-scale, very concurrent environment, and so dealing with concurrency, locks, and other issues like that is part of our daily existence. I'm going to ask you some questions related to that kind of work."

The idea is to give the candidate every chance to shine, and to help them know what to expect — with as much detail as possible — at every step. Keep the interview focused on those things that matter most to the job, and make it clear why those things matter to the job.



Designing Interview Questions That Avoid Creating a Monoculture

As stated previously, an intrinsic part of human psychology is that we tend to prefer people who are like us. Humans have an immense capacity for othering people. I'm safer with my tribe, our brains are constantly telling us. But intellectually, we know that having a diversity of backgrounds, perspectives, opinions, and approaches are better for business. Those differences help us identify new opportunities, new ways of working, and are how we "disrupt ourselves," learn, and grow.

There is an inherent problem with interview questions that seek to focus on "someone's approach to the problem," which is that, unless we're very, very careful, we will bias toward people who use approaches we like and are already comfortable with. That's the main ingredient in building a monoculture, meaning a team that tends to all think alike, all approach problems the same way, and so forth. It is, in other words, the opposite of diversity.
Essentially, almost anything you can think of that's really difficult to create an objective scoring rubric for is also something that can admit conscious and unconscious bias, and those are how you create and sustain a monoculture.

Think about it like this: It's not hard to give someone an Analysis-level exercise that can be objectively scored. For example:

> *Take a look at this router configuration file. Do you see anything here that would concern you from a security perspective? What would you do to fix it?*

+1pt	Candidate identifies the insecure configuration of EIGRP.
+1pt	Candidate explains the risks associated with this configuration (ask the candidate to explain the risks if they don't do so on their own). Include example answer that lists the key points.
+1pt	Candidate suggests enabling message authentication mode.
+1pt	Candidate explains how message authentication mode mitigates the risks (ask the candidate to explain if they don't do so). Include example answer that lists the key points.

This kind of question and rubric doesn't leave a lot of room for interpretation. Now, this question also isn't telling us much about how a candidate mentally works through the problem — how they break down the configuration file in their mind, how they prioritize different areas to consider, and so forth. And while "how someone works the problem" can be predictive of job success, it can also be an area where our biases come into play and lead us to support a monoculture.

It comes down to this: **when you focus on observable job skills, by using an objective rubric, you will make it hardest for bias to creep in.**

So how can you still include the admittedly valuable "seeing how they think" without also inadvertently creating a monoculture?
Consider "pulling back" a little. For example, make sure your interview includes Analysis-level, skill-based questions that can be graded with an objective scoring rubric. Obtain the majority of your interview scores from those. Then, go ahead and include a problem where you ask the candidate to work through a problem — but don't grade them on how they work through the problem, and instead focus on whether they work through the problem. In other words, it isn't the results that count, and it isn't even the "journey" that counts; what counts is the fact that the candidate is indeed on a journey. You can see them working through the problem, even if you don't love the *way* they're working through the problem. For example:

0pt	Candidate struggles to articulate the essential problem, even when prompted.
+1pt	Candidate articulates the problem, but has no suggestions for how they'd proceed next.
+2pt	Candidate articulates the problem, and has a suggestion on where they'd start trying to solve it.
+1pt	Candidate asks follow-up questions to better understand the scope/nature of the problem.
+1pt	Candidate asks follow-up questions related to the business logic they're seeing.

Creating Accountability

Lastly, it's important to hold yourself accountable for your goals — to measure yourself, and to do so with rigor and honesty. The exact metrics you choose will obviously depend on your specific goals, but some good metrics to consider are:

Passthrough ratios. Each stage of your interview should be passing along a reliable portion of your candidates, and should have a meaningful and desirable effect on your overall hiring pipeline. For example, an interview that routinely passes along 80% of candidates is probably not a useful interview. It isn't narrowing your pipeline much. On the other hand, an interview that passes only 20% of candidates is also probably not useful unless it's a very late-stage interview intended to whittle down just a few remaining candidates.

Candidate impact. Ideally, each stage of your interview process should pass along the same percentages of men as women, of people of color as white people, and so forth. You should watch this metric carefully, because less-than-equal passthrough ratios may be a sign that you've designed interviews that will lead to an undesirable monoculture.

Candidate sentiment. Wait, does it matter what candidates think? Absolutely yes. Remember, your interview process is the first time you're meeting a new colleague — hopefully, you want them to come away with a positive impression. If nothing else, word about your organization's interview process will be on the Internet. A poor overall level of candidate sentiment will damage your employer brand and make it difficult to attract new candidates. Some sentiment questions to consider:

- Were the questions we asked clear and easy to understand?
- Were you confident that you understood how you were being scored?
- Were the questions we asked clearly related to the job role you applied for?
- Did the questions we asked seem aligned to the job level you expected?

Job performance. This is a measurement many organizations fail to track, often because once a candidate is hired they pass out of the Talent Acquisition organization and into a production team — breaking the "link" between interview and performance. But this is the most important metric, because it tells you how predictive your interviews are. Candidates who score well throughout should perform well after their first 90 days, six months, and year on the job. If you're not seeing a close correlation between interview scores and ultimate performance, then your interviews need some redesigning.

Now, there's an important caveat about all of this that keeps interviewing from being truly valid science — and it's a caveat you can't do anything about. But it's important to keep in mind, because without this caveat firmly in mind you can become overconfident in your interview approach.

When you conduct an interview and make a decision about a candidate, you are implicitly stating a scientific hypothesis: *This interview does a good job of identifying qualified candidates, and a good job of identifying unqualified candidates.*

In science, a good, rigorous hypothesis exhibit's Karl Popper's basic scientific principle: falsifiability. Falsifiability is the assertion that for any hypothesis to have credence, it must be inherently disprovable before it can become accepted as a scientific hypothesis or theory.

💡	A candidate scored well, and we hired them, and they performed well on the job.	*This proves the positive side of the hypothesis.*
💡	A candidate did not score well, and we hired them, and they performed poorly on the job.	*This is the falsifiable part of the hypothesis.*

Real science has to do both, but obviously as a hiring organization you're never going to even consider the second one. And that's where it becomes easy to get overconfident about your interviewing approach. To put it another way, you'll never know if you left qualified candidates behind, because once your interview filtered them out, you never performed additional tests to see if the interview was correct in doing so or not.

In other words, our hypothesis that our interviews do a good job is inherently unfalsifiable, which means it isn't truly "good science."

As stated, you're never going to be able to control for that, but you do have to keep it in mind. Never assume that just because your interview only passes on good candidates that it isn't also denying some good candidates a further opportunity. This is where monoculture comes strongly into play: a good interview — meaning, one that only passes through qualified candidates — can still create a monoculture, which shouldn't be desirable. That's why we always have to approach our interview metrics with a healthy awareness that we'll often never know if we're doing a less-than-excellent job.

Conclusion

Interview questions have the potential to help you identify highly talented, qualified candidates who will succeed in the role, but they also have the potential to accidentally keep those same candidates out. Fortunately, creating effective interview questions that lead to successful technical interviews can be broken down into several steps:

- Establish what success looks like in the role by performing a job task analysis and then create your interview questions around the job-related tasks.

- Having an interview that is structured, using consistent questions, focusing questions solely on job tasks, and creating a rubric to score answers objectively will help ensure that the interview is fair and free of bias.

- Leave candidates with a great impression and allow them to perform at their best by telling candidates how interview questions connect to the job.

- Even when following the framework provided here to create predictive, fair, and enjoyable interview questions, companies should measure their interviews in order to truly know whether you're passing through qualified candidates without denying other equally qualified candidates.

karat^

Work With Karat to Make Your Technical Interviews More Effective

As the leader in technical interviewing, Karat has conducted over 350,000 technical interviews for companies across the world. When companies partner with Karat, they get access to Karat's innovative, equitable top-of-funnel technical assessment; 24/7, on-demand technical interviews; and the company's expertise and data-driven processes. Whether you're looking to increase your hiring yield, reach more diverse candidates, or rapidly scale your hiring, Karat can help you transform your technical interviews to meet your goals.

Request a demo today ↗