

How To Conduct Effective Technical Interviews



The people who design, construct, and maintain the buildings we live in and work from typically must be licensed to practice their trades, often obtaining said licenses only after grueling examinations. However, the people who design, construct, and maintain the systems and software that run nearly every other aspect of our lives face no similar assessment of their qualifications—other than vendor-specific certifications, perhaps.

No assessment, that is, until they meet the person or team tasked with conducting an organization's technical interviews, interviews designed to determine whether a candidate has the tech chops and skills needed for an open IT position.

Have you been tasked with performing technical interviews? If so, are you up to the challenge? *Are you sure about that?*

You will be on trial as much as the candidates. There are no ifs, thens, or buts about it: you'll be judged during technical interviews, too. When structured properly and performed effectively, they become dialogues with candidates whose talents will fill a much-needed void, bring the skills needed to launch a new project, improve an organization's overall success, complement the work of an existing team—or all of the above.

Technical interviews can also be one of the best opportunities to pull back the curtain and show IT candidates why your organization is the one they should most want to work for.

Keep in mind that exactly the opposite could happen if you don't have a thorough, carefully thought-out strategy that guides the questions you ask candidates, influences the way you ask them, and focuses your expectations on the right outcomes.

Don't be part of the problem. Use the tips and tactics that follow to learn how to conduct effective technical interviews, and do so in a manner that engages candidates instead of instilling fear or completely alienating them.

What not to do

Gamblers and daredevils can do what they do "at their own risk." But as a recruiter, HR professional, or IT manager, you have more than just yourself and your own reputation at stake. Your organization or the company you represent will ultimately suffer should you overestimate your technical expertise and your ability to measure it in candidates.

Filling These Positions? Test Them!

Application Architect
Back End Developer
BI Manager
Business Analyst
Computer Programmer
Computer Support Specialist
Computer Systems Administrator
Computer Systems Analyst
Data Warehouse Manager
Database Administrator
Front End Software Developer
Information Security Analyst
IT Director
IT Manager
Java Developer
MS SQL Developer
.NET Developer
Network Architect
Network Engineer
Project Manager
Security Analyst
Software Developer
Software Engineer
Systems Administrator

If you're not a technical person, do not conduct a technical interview on your own. Period. Just don't do it.

Even if you consider yourself tech savvy, step back for a moment to objectively evaluate what you do know versus what you don't know. Interviewing candidates for soft skills is one thing, but assessing the competencies of seasoned tech professionals is another.

.

If you can't recognize greatness, how will you hire great candidates?

Don't cross the Grand Canyon without a safety net. It's lonely at the bottom. Unless you're able to intelligently and confidently discuss the nerdiest, nittiest, and grittiest of the technologies required for a position, then bring in help. Better yet, let someone else in your organization conduct technical interviews altogether.

If you nevertheless decide to go it alone, you may be tripped up by one of these three common reasons why technical interviews fail:

1. You have a job description in one hand, and a candidate's resume in the other. They seem like a perfect fit—at least all the acronyms match! But do you know enough to question the claims made by both? If you don't, you risk on-boarding candidates who look well enough on paper, but quickly fail to meet the demands of a position and the expectations of colleagues.
2. The best candidates are good at what they do. More than that, they're good at figuring out what they'll need to learn to be good not only today or tomorrow, but next month and next year. *Are you* good enough to ask the questions that will identify the candidates who will be rock-solid starters *and* continue to grow?
3. Your lack of knowledge could lead to an interviewer's worst nightmare: Instead of identifying (and subsequently hiring) the most qualified candidates, you instead alert them to make a beeline for the door—and perhaps to seek employment with a competitor.

.

Screen candidates for the ability to learn emerging technologies...unless your organization is happy with the projects, networks, and systems it has today.

Especially in highly competitive IT disciplines, it's a candidate's market. And a good candidate can sense when he or she is the smartest person in the room. If you fail to make a good first impression through intellectually engaging and compelling interview questions, the best candidates may feel slighted—and never look back.

Knowing when an answer is right or wrong isn't enough. You must be able to sift through poor, acceptable, good, great, and "why hasn't our company ever thought of this before" ideas.

Finally, no matter how smart and informed you are, if you're not considerate of your candidates during the process of their technical interviews...they're not going to leave with a good feeling about the position or about your organization.

Instead, let applicants know up front how many steps and how much time will be involved in your interview process, like in this example:

Expect fifteen minutes to complete an online test. Which, if successful, will be followed by a 30-minute phone call. Which, if successful, may be followed by an hour-long, on-site technical test...

But don't worry, it's not designed to be finished. What we want to see is how you approach and solve problems. When you're done, we might discuss why you did what you did for about thirty minutes. We may ask you to talk to other members of the team, and we'll try to let you know if we think these meetings will run longer than a couple of hours. And sometimes we assign homework to check how you'd use your knowledge to solve a specific problem...but we try to keep it short and simple.

Be transparent with candidates. When you give candidates the opportunity to choose whether they think it's worth it to continue to participate in your technical interview process, you will ultimately find the ones you're looking for.

Mix and match

Once you've decided who will conduct a technical interview, it's time to determine what to ask, and how to ask it. You'll want to identify processes you can rely upon, processes that can be repeated and measured with the same time limits for all candidates.

GET STARTED BY CONSIDERING THESE METHODS:

A SIMPLE ELIMINATION QUESTION • MULTIPLE-CHOICE QUESTIONS
CODE SAMPLES • BRAINTEASERS AND GUESSTIMATES • WHITEBOARD CHALLENGE
TIMED ONSITE CODING/ARCHITECTURE/ANALYSIS TEST
WALKTHROUGH PREVIOUS OR PERSONAL PROJECT
PAIRED PROGRAMMING OR PROBLEM SOLVING • HOMEWORK

Each method has strengths and/or weaknesses. To tailor a technical interview to meet the needs of a specific position, team, or organization, you'll likely end up defining a protocol that combines a number of them.

A simple elimination question. Some organizations require applicants to answer a simple question before they will accept resumes. Want to work for Company A as a software engineer? You

must pass *this* first. Are you a project manager who sees a huge market opportunity for Company Z? You'll be invited to send your resume after successfully completing *that* test. For example, an organization looking to hire an SQL developer might ask all applicants a basic database question like, "Write a query that returns the number of customers with the last name Smith."

PROS	CONS
<ul style="list-style-type: none">+ Reduces number of resumes to be screened.+ Failures eliminate most who lack fundamental skills.+ Passes include those who lack "required" education or experience.+ Online platforms provide simple tests across many disciplines.	<ul style="list-style-type: none">- The question's very simplicity might offend seasoned professionals.

Multiple-choice questions. Organizations use MCQs, as they're also called, when filling positions or hiring in industries where knowing "right" and "wrong" information is vital.

PROS	CONS
<ul style="list-style-type: none">+ Useful when specific working knowledge is required.+ A good way to measure analytic abilities.	<ul style="list-style-type: none">- Can be tedious, and because questions are often shared online, can be easy to prepare for.- MCQs that test generic knowledge of a discipline don't assess a candidate's ability to meet an organization's current challenges.

Code samples. One way to see how coders code is to ask them to provide samples for you to evaluate. If you want to *really* impress a developer, check him out on GitHub first. Thoroughly look at his depositories, sources and forked code, and public activity—then ask him to discuss his choices during your interview.

PROS	CONS
<ul style="list-style-type: none">+ Emailed samples allow evaluation before, and review during, interviews.	<ul style="list-style-type: none">- Possible IP or NDA issues if an applicant provides samples written for a current or former employer.- Easy to get help preparing sample, so candidates must explain code.

Brainteasers and guesstimates. Brainteaser questions like the classic, “Why is a manhole cover round?” generally don’t have a right or wrong answer, but are asked because they provide insight into an applicant’s ability to think on his feet while under pressure. Guesstimate questions like, “How much does the Golden Gate Bridge weigh?” may or may not have a correct answer, and are used to assess how a candidate approaches a question and how she ultimately reaches a conclusion. If you plan to ask these types of questions, search the Internet first to see whether creative answers are readily available on sites like Glassdoor or Quora—if you can find a plethora of answers, so can the people you will be interviewing.

PROS	CONS
<ul style="list-style-type: none">+ Easy way to observe problem solving, creativity, and composure.	<ul style="list-style-type: none">- Candidates can prep for many questions/solutions in advance, or post the question online after an interview for others to see.

Whiteboard challenge. Though most candidates dread whiteboard questions (or similar written challenges that substitute paper and pen or a laptop), having them write out a solution while explaining it is a common practice in IT and other technical fields. It shows whether a candidate knows how to solve the problem, and more importantly, how he goes about reaching a solution. It also allows you to evaluate an applicant’s communication skills and composure.

PROS	CONS
<ul style="list-style-type: none">+ Can be used to evaluate candidates in any tech discipline.+ A great method to follow a candidate’s thinking process step-by-step.	<ul style="list-style-type: none">- Whiteboard-style questions take time, often adding 30–60 minutes to an interview.- Top respondents may simply have had more practice with this type of challenge; easy to favor talkers over doers.- Left-handed candidates may smudge answers on whiteboards.- Unqualified interviewers can misinterpret a candidate’s work.

Timed onsite coding/architecture/analysis test. A timed test can be used instead of a whiteboard question. The methods are similar—candidates are given a set of questions to answer within a certain time frame. Here, however, they aren’t asked to write out and communicate their solution on the fly, in front of an interviewer or team. Instead, they’re taken to a quiet area and left alone to complete the test. When the time for the test expires, ask candidates to walk you through their answers so you can evaluate their approach to each problem.

PROS

- + Can be used to evaluate candidates in any tech discipline.
- + A great method to follow a candidate's thinking process step-by-step.
- + Increasing the difficulty level of questions throughout the test helps filter candidates who don't know an answer or have spent too much time over thinking a solution.

CONS

- It's time consuming. Be sure to include review time in addition to the testing period.
- Unlike with a whiteboard question, you won't get to see candidates thinking on their feet.

Walk through a previous or personal project. Perhaps the best way to approach this question is to ask a candidate to walk you through the project they are most proud of. It doesn't matter whether it's something they did for an employer, as a pet project, as a contribution to an open source project, or even while in school. If the project is one that was developed by a team, ask questions to determine the applicant's personal contributions to its success.

PROS

- + Can be used to evaluate candidates in any tech discipline.
- + A great method to follow a candidate's thinking process step-by-step.
- + Reduces pressure on candidates and gives them an opportunity to shine.

CONS

- Easy to rehearse, so be prepared to probe.

Paired programming or problem solving. Another alternate to the whiteboard is to sit down with a candidate and work through a problem together, like colleagues do. If using the same question with all candidates, you can easily compare the thought processes and flexibility behind each one's solution.

PROS

- + Gives a real feel for candidate's style, communication skills, and approach to teamwork.
- + Can be used to evaluate candidates in any tech discipline.

CONS

- More difficult to measure and record results.

Homework. Give promising candidates a "homework" assignment that tests for a specific skill or skills critical to a position. Try to design questions so the test will not take too much time for candidates to complete—perhaps a couple of hours—and let them know upfront how much time you

anticipate the question should take to answer. Give candidates a reasonable amount of time to complete the assignment. Upon submission, review the answer. If it is favorable, be sure to review it with the candidate personally to confirm the work is theirs.

PROS

- + Asking every candidate the same question makes it easy to compare and rank results.

CONS

- Candidates may consider this an attempt to get them to do “free work,” so be sure the assignment can’t be construed as work that will benefit your company.
- Candidates may find it difficult to make time for homework.

Testing, testing...1-2-3

Some argue that technical or coding tests during interviews don’t accurately convey a candidate’s abilities, or that it’s just plain unfair to expect a stellar performance under the already stressful environment of a job interview. But doesn’t your organization need people who don’t crack under pressure? Who are quick-thinking problem solvers?

Technical tests are simply the most effective way to determine whether an applicant has the specific skills needed for a position.

Once you’ve reviewed the above methods used to conduct technical interviews and decided on a mix of techniques that will work best for your organization, add one or more of the following strategies to help identify the most suitable candidates.

Ask questions you know the answers to. Yes, this is a no-brainer. It has an added benefit, though. When you ask a question you’ve seen solved many times, you’ll be able to quickly assess which responses are “good” or “bad” because of your familiarity with them.

Ask a simple question. Don’t overlook the usefulness of easy questions. How people answer them can reveal much about how they initially approach any problem, large or small. Is their first instinct to solve it quickly? Could their solution be more elegant?

Ask for more than one answer. See whether an applicant can compare approaches and identify the constraints and tradeoffs of different solutions.

Ask questions that get harder and harder. Let candidates know you’re not looking for a perfect score, or even for them to finish a set of questions. You’ll be able to compare candidates’ skill levels when reviewing answers. Look for not just right/wrong, but for approaches that will gel with the role to be filled.

Ask a question that requires a stretch. Test flexibility and intellect by asking applicants something they know little or nothing about, and take note of whether they become frustrated or are comfortable pushing past the edges of their knowledge. Nearly every IT role benefits from the ability to take on and solve complicated problems.

Ask this question to test communication skills. Have candidates answer a question in their area of specialization, one they should be able to answer with confidence. Then ask them the same question again, but have them answer as if you were only ten years old.

Ask a question involving someone else's code. Coding candidates should understand code written by someone else. Ask them to explain a sample of code. If they can, press for additional knowledge by asking whether they can see a way to simplify it, or request they add a feature. A similar exercise would be to provide code that's missing a feature, and ask candidates to fill in the blanks.

Ask a debugging question. Because debugging skills are often just as important as coding ability, provide developers with a sample source to debug.

Proper documentation

No matter how comprehensive and targeted your technical interviews are structured, unless you document each stage of the process, you will likely be unable to recall why you thought a candidate was great rather than merely good.

Don't rely on your memory. Once you've decided on the best methods and questions to include in a technical interview, figure out a way to rate or score each candidate's performance during each portion of the interview. Take detailed notes during the interviews (did a candidate require a nudge in the right direction, become flustered, or show excitement), and include copies of all the written work a candidate completes. (If using a whiteboard, take a picture!)

You must be able to capture your interview observations as fully as possible so they can be shared—and hiring decisions made—accompanied by records that document your recommendations.

Keep in mind that while you are testing a candidate's technical skills, he or she is evaluating you and what they see of the organization. Use this opportunity to sell them on the benefits of working there, and be sure candidates get a good feel for its culture.

Finally, work on ways to quickly move candidates through your interview pipeline. Be respectful of their time, update them of any changes or additional requirements, and if a superstar shines in an interview, don't hesitate to escalate the on-boarding process.

Putting it into action

Still unsure about how to conduct effective technical interviews? ExpertHiring makes it easier to get started. We specialize in quickly filling the most difficult IT openings, and we're here to help you do the same.

Don't wait! Click to download our list of [technical interview questions](#) today. They'll prepare you to interview a cross section of the most common IT roles our clients—and likely you—need to fill.

Hiring for more specialized positions? Call or email ExpertHiring to see how we can help identify the candidates with the technical skills you need.



1-800-385-6133 (US/Canada) or 0800 368 8699 (UK)

Email: HelpMeHire@ExpertHiring.com

Web: www.ExpertHiring.com