

Are code tests blocking 1 in 3 developer hires?

May 21, 2020



Data shows that a systemic flaw in code tests can reduce hiring yield by nearly 1/3

A few weeks ago we connected with Claus Moberg, Roblox’s VP of Engineering, to get his thoughts on the state of hiring software engineers. One theme that came up is the impact of false positives. He noted that live first-round technical interviews showed “...a dramatic improvement over [code tests] and our own emailed questions. Without a live technical interview, too many false positives get through, causing bad onsite and impacting team morale.”

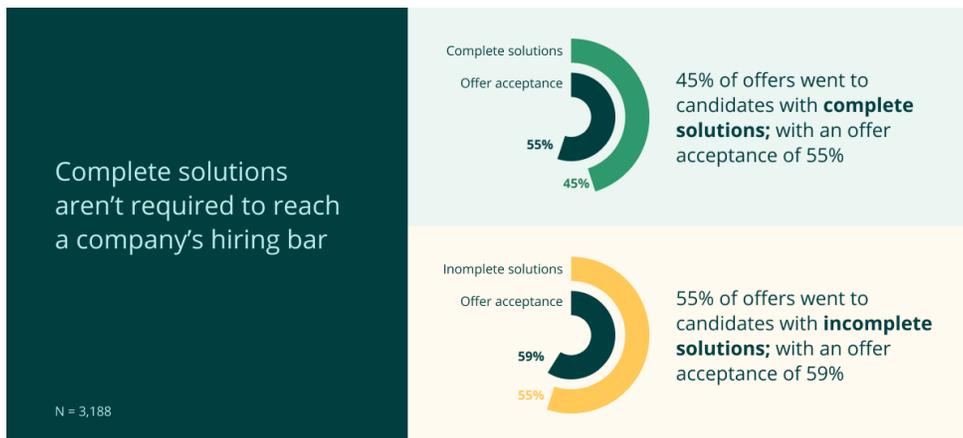
Starting every hiring process with a live technical interview might seem like a luxury in a world where software engineering time is severely constrained and the financial outcomes of product development can have a seismic business impact. However, due to poor candidate experience and poor signal compared to live interviews, [code tests](#) often end up taking more engineering time to vet candidates and decrease hiring yield from an already limited talent pipeline.

One reason for the poor signal produced by code tests is the requirement that candidates’ solutions be both complete and correct in order to “pass.” These are critical drivers of signal but, requiring perfection filters out too many candidates, and as Moberg pointed out, often filters the wrong ones in. On one hand, this is unfair to candidates and on the other, costly to recruiting efforts.

55% of offers go to candidates with incomplete solutions

A code test requires completeness simply by its nature. It requires a candidate to write and then run a solution to a single problem. However, failure to complete the first question doesn’t mean 100% failure to reach the company’s hiring bar.

In a live interview, an experienced [interviewer](#) will be able to get sufficient signal to provide an accurate recommendation regardless of completeness. Reliance on completeness is in fact so unimportant that Karat data show 55% of offers go to candidates with “incomplete” solutions, and they accept them 59% of the time.



More than half of total offers were extended to candidates with incomplete solutions and their acceptance rate was on par with candidates who provided complete solutions.

12% of offers go to candidates who received guidance from an Interview Engineer

Correct answers do matter, but how a candidate gets there also matters. Code tests often require bulletproof correctness, which the candidate has to get to on their own. This is in contrast to a real team environment in which software engineers work together to solve big problems. They'll use code review, quality assurance, and other checkpoints with team members to ensure their code will work as intended. Most are not Bertram Guilfoyle.

Interviews should mirror this environment. Signal on critical competencies increases when a candidate can ask the interviewer questions to check that they're on the right track. Likewise, an interviewer may want to give guidance to a candidate who has overlooked something as simple as a typo.



Among candidates who received offers, 49% of those who received minor guidance accepted, but 75% of those who received significant guidance accepted the offer.

Giving guidance doesn't mean giving away the interview. It can impact the recommendation following a live interview and acceptance rates by highlighting opportunities for signal, reducing anxiety for the candidate (thus improving their performance), and demonstrating collaboration.

Absolute completeness and correctness come at a cost

Surfacing signal where it may otherwise have been missed has a massive impact on hiring yield — in particular at the middle and bottom of the hiring funnel where the most investment has been made. The data above suggests swaths of candidates could be getting filtered out of the hiring process due to code tests that over-index on completeness and correctness when higher value signals are seen in a live interview.

The binary pass/fail inefficiency of code tests is a systemic problem because it immediately eliminates all of your close-but-not-quite-there candidates from the hiring process. In a hiring process with a live first-round technical interview 55% of all offers go to this contingent, and those candidates accept at a 59% rate. This means for every 100 software engineering hires made as the result of a process that starts with a live interview, 32 of them could have been lost in the code test's requirement of absolute completeness and correctness. That's a steep price to pay.

Humans are complicated things. Navigating emotional cues and innocuous missteps to find indicators of skill is something humans are uniquely qualified to do. Giving experienced developers the right technology and best practices to administer a live technical interview filters in large numbers of candidates. This is, in a nutshell, why Karat created the job and discipline of Interview Engineering. Without it, teams would miss out on great developers, and candidates would miss out on great job opportunities. The data says it loud and clear: What's good for your candidates is also good for your company.