**The Technical Interview Is Dead (And No One Should Mourn)**

Allow me just a little self-congratulation. Two years ago I wrote “[**Why The New Guy Can’t Code**](http://techcrunch.com/2011/05/07/why-the-new-guy-cant-code/),” about my contempt for the standard industry interview procedure for software engineers, condemning Microsoft and Google in particular for their [**brain-teasing**](http://online.wsj.com/article/SB10001424052970204552304577112522982505222.html) [**riddles**](http://www.businessinsider.com/15-google-interview-questions-that-will-make-you-feel-stupid-2010-11) and [**binary search**](http://ecarmi.org/writing/google-internship/) questions. And lo and behold, this week Google’s head of HR [**admitted**](http://www.nytimes.com/2013/06/20/business/in-head-hunting-big-data-may-not-be-such-a-big-deal.html): “Brainteasers are a complete waste of time.”

But wait. Let’s just unpack that interview a little further:

*GPA’s are worthless as a criteria for hiring, and test scores are worthless … Your ability to perform at Google is completely unrelated to how you performed when you were in school, because the skills you required in college are very different … Academic environments are artificial environments. People who succeed there are sort of finely trained, they’re conditioned to succeed in that environment*

You know what else is an artificial environment that doesn’t translate at all well to the real world? That’s right: *the entire technical interview process*, as traditionally performed. It’s not just brainteasers: it’s the fundamental concept of being brought into a room, grilled on the spot with technical questions that must be answered without any of the usual resources, and then being made to write code on a whiteboard. All this on the nonsensical pretext that it’s a decent measure of whether the candidate is a good software engineer.

I’m lucky: I’ve always been quite good at this arbitrary and ridiculous skill — but at the same time, its requirement has always seemed so obviously *wrong* to me. Nowadays, though, finally, more and more voices are being raised against this madness. I give you [**Jeff Atwood**](http://www.codinghorror.com/blog/2012/03/how-to-hire-a-programmer.html), **[Udo Schroeter](http://devinterviews.pen.io/" \t "_blank)**, [**Nelly Yusupova**](http://www.entrepreneur.com/article/225932), [**Josef Assad**](http://www.josefassad.com/on_hiring_good_engineers), and especially [**Treehouse’s recent post**](http://wellbredgrapefruit.com/blog/2013/06/01/how-we-hire-developers-at-treehouse/) on how to hire developers. Oh, and, er, me, [**a year ago**](http://techcrunch.com/2012/05/12/bloodsport/), saying much the same thing.

Many of the comments on that year-ago post were very angry. Nobody likes to be told that they’ve been screwing up the most important part of their business (and recruitment is almost certainly the most important part of your business) since Day One. But the sooner people realize that the traditional “come in, answer random quiz questions, and write code on a whiteboard” model is horribly broken, the better off they’ll be. If you click through to and read the above, you’ll find that they’re mostly minor variations of a much-improved process:

* Quickly filter out the technically inept by asking half a dozen basic technical questions – Atwood calls this “[**the FizzBuzz filter**](http://www.codinghorror.com/blog/2007/02/why-cant-programmers-program.html).” Ideally you can do this online. You’ll be amazed how many people fail. (If they’ve been recommended to you, you can skip this step.)
* Talk to them–in person if they’re local, voice-only if they’re not–about technical problems they’ve faced, tools they use, decisions they’ve made, pet peeves, etc. This is the “behavioral interviewing” that Google’s data shows to be actually useful. Note, however, that while you’re discussing technical concepts, you are *not* grilling them with technical questions that must be answered correctly.
* Above all, discuss their past projects, how they got them done, and the decisions they made en route. Maybe have them talk you through some of their code on Github. To reiterate my own line: *Don’t hire anyone who hasn’t accomplished anything. Ever.* If a developer doesn’t have a portfolio they can talk to you about, not even any side or pet projects…then don’t waste your time talking to them at all.
* Try to establish “cultural fit” … although be very careful that you’re just talking about *work* culture, e.g. enterprise vs. startup, and that this doesn’t lead to “subconsciously excluding people who aren’t just like you.” Don’t become a frat full of brogrammers.
* Finally, if they’ve gotten this far, give them an audition project. Something relatively bite-sized, self-contained, and off-critical-path, but a real project, one that will actually ship if successful. Hire them on a paid basis for a week or so to build it, and keep a close eye on their code and progress. (If you do pair programming, have them pair with your existing team.)
* Hire them. Or don’t.

This process is better for everyone. Employers learn far, *far* more about potential employees; waste less time in interviews; and get them an excuse to pay people to do all those little beneficial things that would otherwise be back-burnered indefinitely. The only real problem is that it doesn’t scale. Google, for instance, would need to find literally *thousands* of one-week audition projects every year for its candidates, which is probably not realistic.

The process has to be attractive to developers, too, in this seller’s market. (I sometimes wonder how many good engineers have been put off by Google’s previous, rather insulting, hoop-jumping interview process; me and most of my techie friends, for starters, but that’s probably a self-selected group.) Fortunately, “better than the old way” is a very low bar, and most good developers casting about for a new gig would far rather work on a real paid interesting project for a week than spend a day regurgitating code on a whiteboard.

My headline here is more aspirational than descriptive, I admit: the technical interview isn’t dead *yet*…but it should be, and soon enough it will be. What baffles me most is why it’s taken so long. The whole purpose of an interview was to serve as a proxy for actual performance, because we didn’t have the tools and infrastructure to easily observe and measure the latter: but now that we *do*, it is the height of cargo-cult stupidity not to use them. Stop quizzing people, and start finding out what they can actually do.

Mikhail Jacques

2 minutes ago

I most certainly agree with the article. In the modern world of digital divide, those who have access to the ocean of digitized data and know how to extract the information they need, need NOT memorize stuff. The new paradigm of learning is to replace an empty mind with an open one. Asking an interviewee to write a piece of code on a piece of paper is as stupid as it gets. One cannot possibly know everything by heart. But one should know how to learn, how to understand and how to get what one needs from the vast ocean of digital data!

JoeCascio

29 minutes ago

This is so true. I've always felt that if you can't tell I'm competent just by talking to me, then you're not competent enough to interview me. I am very glad to have never had to endure any stupid brainteasers or ambush questions and when I was on the other side of the table, I never asked anyone I was interviewing to endure them either. I told people what we were doing and just let their reactions wash over me, following up on the more interesting ones. I asked open-ended questions and basically just followed my nose. It's usually really easy to tell when someone's putting spin on answers or trying to inflate their experience.

randunel

3 hours ago

In response to "GPA’s are worthless as a criteria for hiring, and test scores are worthless … Your ability to perform at Google is completely unrelated to how you performed when you were in school, because the skills you required in college are very different … Academic environments are artificial environments. People who succeed there are sort of finely trained, they’re conditioned to succeed in that environment"

A good GPA and good academic results aren't totally worthless. It shows that the candidate is responsible. After high-school and the irresponsible teenager, the candidate succeeded in the academic environment. Some companies may be looking for responsibility when checking the academic results, besides the abilities to learn and adapt to another environment.