

Positioning

Q: What is our product doing?

A: Earlier this year we were wrapping up the C-round of Y-Comb. Now that we have capital, we hire. We were hiring mostly Lean 4 devs, but also Rust. The automated Rust test and grading saved us a lot of time. The real advantage was to be able to compare and know how to compare.

Some of the candidates interviewed well, some didn't. We were a small team and we wouldn't be able to rank them objectively. Being able to look at people's results in a more standardized way caused us to not make an offer to a person who did well in the interview, but did poorly in the test.

Conversely, zerohr helped us make an offer to someone who didn't do well during the interview, was shy and nervous, but they blew the test out of the water and ended up being an incredible contributor to our entire project.

I think about what would have happened if we made an offer to the 1st person rather than the 2nd, and I think that we would've been very behind in our project, because making a wrong hire is very expensive. Not making the right hire is also very expensive in the opportunity cost.

Q: Do I understand correctly that in your opinion, our project is giving you a way to objectively rank the developers?

A: Correct, and also do it in a more automated way so that evaluation of tech skills is not solely on the judgment of the person making the evaluation.

Q: If zhr wasn't a thing, what would you do?

A: Probably what I did in the previous companies: interview + take home git repo of lower sophistication than zhr. I would put 20-30 hours into making such a repo for some other companies I worked with. I think that zhr allows us to get this because instead of 30 hours, it's 100s of hours of a team thinking about what it means to be a good rust developer. Also, a good advantage for us, it allows us to not hire people based on aesthetics which I would encode.

Another thing is that we were hiring a total of 8 people, but I think that zhr would scale really well if we would be hiring twenty or thirty or more. I intend to use zhr moving forward.

Q: You said that you'd roll your own thing. Does it mean that you feel like we don't have competitors? If not, then who do you feel our competitors are?

A: When I think of competitors, I think about triplebyte <https://triplebyte.com/company> and conventional recruiters. Most other platforms are focused on sourcing and asking for a big chunk of their salary. We don't need this stuff though, we're capable of generating 10x candidates than we have positions. It's not worth our money to spend on it.

Zhr is doing something specific for us: you have a bunch of candidates, you want to make hires. How can we give you information that will help you make the best possible hires from that set of candidates. That's what we needed. That's what we got. I can't think of any other service that would give a centralised rust technical test to the candidates we provided and give us the results, and let us make decision.

Q: Some people compare us to hackerrank. How do you feel about that comparison? What are the differences?

A: *looks up*. The issue I have with hackerrank, code kata and exercism is that to be good there, it's necessary and sufficient to be good at solving CS problem sets. In the industry, even R&D, stuff like big-O doesn't matter much because it's a collective process. It's not my scene. My scene is people who appreciate types. And I don't see hackerrank caring about types in any capacity.

Testimonial proper

Q: What would you say was the main obstacle that could prevent you from using our offer?

A: Personal connections, trust in the sales person (Jonn). We don't just make our hiring decisions based on what ZHR says. The trust here is not whether or not you trust someone else to make a hiring decision for you, the trust here is whether or not this information puts you in a better position to make a good hiring decision. I looked pretty carefully at the coding task, evaluation process, materials around that like instructions. And I was able to, within an hour, verify myself that this is at least as good if not better than what I would come up with after 20-30 hours. And I wouldn't make any out of automation stuff.

Note: our automation also check for statistical stability of the results, which assures the signal strength.

A: And it will become even stronger as you have more data.

Q: We're skipping the question "what did you find as a result of buying ZHR".

A: Ok.

Q: Name the single most important feature of service or the automation package and the service that you like the most?

A: The packaging

1. PDF,
2. git repo,
3. send it to candidates,
4. candidate fills out stuff,
 - a. It's all clearly labeled
5. ZHR runs it trough the tester
6. The result is clearly expressed set of outputs saying
 - a. "this candidate passed these tests, this candidate didn't pass these tests"
 - b. I really liked the names of the tests, they were evocatively named

The structure of the tests as well. The candidate implements some sort of function. And tests are structured like this:

- Our test has a lot of corner cases
- Some ambiguities in the inputs
- And then ZHR sends tons and tons of various inputs and see what outputs the filled in function produces

In a way programmers are making functions from the vague handwavy guidance and this approach models it really well.

Q: If you would have to distill three particular benefits of using the service vs not ranking the candidates vs using a competitor?

A: Let's suppose that we go with a competitor. And let's assume that someone can provide us the signal of the same quality. But since the economics of our competitors is do sourcing first and then asking a chunk of the year's salary. So we've easily saved tens of thousands of dollars per candidate like this. And that's assuming that the signal was as good.

And then the quality of the signal. Suppose it's worse and you hired a wrong candidate. You worked with them for a month and suppose it's not working out. You'd better trust them, because normally IT people respond well to trust to make decisions.

You trust them, and then they go ahead and make bad decisions. And then they contribute technical and organisational debt. That work turned out to not be valuable, and actually negatively valuable, because then it made us spend time of other engineers to undo the bad decisions. This is priceless in some respect.

The converse is hiring someone that wouldn't have as strong of a signal without using ZHR. Especially for a startup, we're playing moneyball (as in the Brad Pitt movie <https://www.youtube.com/watch?v=-4QPVo0Uizc>). **We're not gonna be able to give people as good of a compensation package as, say FAANG, although Facebook is firing a bunch of people, so who knows. There are a lot of developers who are gonna do the economically rational thing and go work for these places. We need to find people who are really motivated by the mission or the philosophy.**

But another advantage for the small companies is that we can get a more detailed look at each candidate and decide to take a chance on people that don't have the background to go and work in the big tech but are as skilled if not more so. And we can give them a shot at proving that.

And that is where ZHR service really excels, especially for the small startups. Because if it can give a signal about these engineers who are on the bubble and it's a good signal and it allows us to hire someone who's really good, well, then it's just gonna create an ENORMOUS value for the company.

I can't imagine how our team would be working without that positive vote from ZHR on these candidates that we ended up hiring. Some of our best people. This one particular guy we hired went from knowing just a little bit of rust to doing surprisingly sophisticated stuff with our codebase. We hired someone junior level, and it turns out they're contributing on the senior level. *Shrugs* moneyball. And perhaps that's a correct framing for it.

ZHR is to technical hiring is what **Sabermetrics is to baseball team management.**

Q: How would you recommend ZHR to other companies?

A: Let me say something critical: ZHR works well only for IT companies that are already doing the right thing. There's some structural analogy between testing a candidate with ZHR vs testing a function with quickcheck or property testing.

So teams and companies that are already inclined to have good practices and are doing the right thing. However, those whom I'd consider braindead, it's not gonna be of much benefit to them. If you're not capable of understanding the signal, then the signal is useless to you. If your company has too many engineers and doesn't deploy, then what good will it make for them to get better engineers? For example, I wouldn't recommend ZHR to Oracle because they don't have a clue of what it means to make good software. They're just gonna spend a huge amount of money on developers, burn them out, and then put that cost on their users who are generally companies and so are relatively invariant to price in a lot of regards. I know a lot of companies like this. That's what I feel is Tesla, and that's something Musk is trying to replicate at Twitter. At least in my opinion, it's the wrong approach to figuring out how to build good software.

Unfortunately I think companies like this are a big chunk of the market. No idea what's the solution there except for evangelizing and perhaps making some middle-ground solution for companies who are ready to be receptive to best practices. So ZHR is really good for us and for our philosophy, but I'm not sure how useful it would be to people who have different ideology.

Q: What would you like to add?

A: Basically, what I'm trying to say is that if you don't like ZHR, you may be too dumb to use it.