

**Hiring with pull  
requests  
...or not**

# Who are we?

**Julik Tarkhanov**

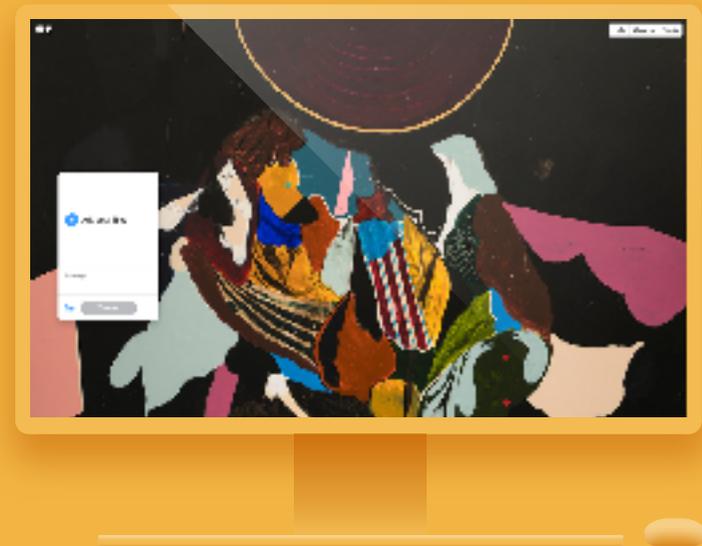
Super Senior Principal Fellow Backend Engineer

**Rebecca Hill**

Super Senior Principal Fellow Frontend Engineer



@WeTransfer



# Approach A

## (Pull request challenge)

- A candidate who does not have enough code available gets asked to do a pull request to an existing open-source library
- The assignment includes having your PR reviewed, in public, and processing feedback on that review

# Objectives

How can the candidate read code

How the candidate is able to integrate their work into the work of others

Is having code out in the open, early, in unfinished form, uncomfortable for candidate

How the candidate handles code review

Assess technical skill level without a technical interview

# **Additional perks**

If the candidate makes architectural suggestions / sweeping design changes

If the candidate debates merits of the stack / wants to prove their convictions by using a different stack

# Benefits for the candidate

See the code+design style of the company you apply to

See how feedback is provided and what the expectations are

Contributing to an open source thing your current employer does not own, control or withhold (your name stays on the commit)

Expanding your portfolio of work you can show to anyone, with no restrictions

Not spec work and not throwaway work

**Example**

# Approach B

## Standalone code challenge

- A candidate who does not have enough code available gets asked to do a take-home code challenge to demonstrate their skills - something small, realistic and relevant to the work they would be doing
- The assignment is then discussed in a technical interview, and an addition/problem fix is made while pairing with a developer from the team

# Objectives

Assess technical skills in a realistic situation

Gives the candidate the ability to pick the tools and approach they are most comfortable with

How the candidate can talk about and explain their code

How the candidate handles code review in person

How the candidate handles pair programming

# **Additional perks**

If the candidate makes an unusual choice of framework that works surprisingly well

Gives the candidate the ability to add an element of surprise or personality to the assignment

# Benefits for the candidate

Opportunity to prove your technical skills in a realistic way (rather than theoretical coding questions)

Ability to prove your skills in a way you are most comfortable with (ie not on a whiteboard, at home, on your own computer, with your own choice of tools)

Can experience what working with developers at the company would actually be like

And what the kind of features you might work on could actually look like

**Example**

# Sequencing is important

Do you do the HR / screening call first or the technical test?

What if the candidate scores very well on one but not on the other?



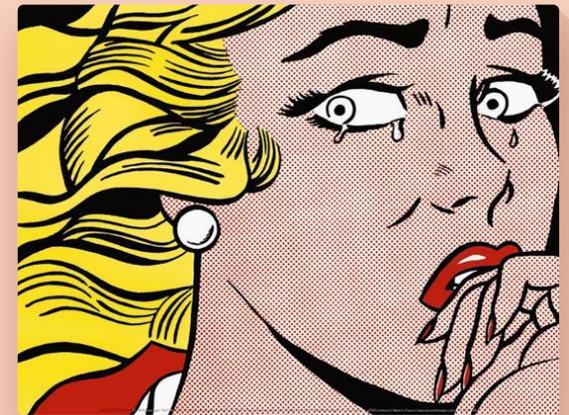
**Sequence?**

**1, 3, 5, 7, ...**

[DontMemorise.com](http://DontMemorise.com)

# Why these can suck

- Both of these approaches can be very intimidating to people - both technically and in regards to the personal assessment of your work
- Doing a code challenge is throwaway work
- Having people do work for free also sucks
- There are no guardrails on the time commitment for either approach
- Confirms your biases towards candidates that favour choices that are consistent with yours
- You want to discern whether the person likes the work, but these are not going to reveal it. A conversation is better.



**Let's debate!**

