Julia Evans

Questions I'm asking in interviews

• favorite • interviewing •

In a fit of "open source your interview process", I tweeted yesterday with the list of questions I'm drawing from when interviewing. A lot of people responded in the gist with amazing suggestions, and I thought I'd consolidate them here so they don't get lost in my pile of gists.

My basic strategy is to spend 20 minutes before each interview I do and pick some appropriate questions from this list. I've tried to categorize them a bit. A lot of these are outright stolen from Edward O'Campo-Gooding's list of questions, as well as from various people at Hacker School. I'd love suggestions for more!

Special thanks to @bmastenbrook, @marcprecipice, @danluu, @kelseyinnis, @zmagg, @graue, and @ircolle for awesome question suggestions.

Edit: A few more things:

- I don't ask all of these in first interviews. Use your discretion, and do what you feel comfortable doing.
- Asking a lot of questions shows that you value yourself and that you're careful when making decisions. It's a good thing.
- If you have an offer, you can schedule extra conversations if you feel that not all of your questions have been answered.
- It can be worth asking the same question to more than one person.

Engineering practices

- What version control system do you use? (if none, the interview should be over =))
- Do you test your code?
- How do you make sure that all code is understood by more than one person?
- Do you do code review? Does all code get reviewed?
- Do you have an issue tracker?
- Describe your deployment process – how do you find bugs in your team's code? What recourse do you have when you find a serious bug in production code?
- Who is responsible for doing deployment? How often do you deploy?
- How do you think about code correctness?
- When something goes wrong, how do you handle it? Do devs get shamed for breaking the build?
- How/when do developers talk to non-developers? Is it easy to talk to the people who are will be using your product?

- Can I see some code the team I'm interviewing for has written? (from an open-source project you work on, for example)
- Who are the people at your company with a lot of depth of experience? Will I be able to talk to them?
- What's your approach to technical debt?

Management style

- How does engineering work get assigned?
- How are technical decisions made and communicated?
- How do you balance support work and feature development?
- Can you give me an example of someone who's been in a technical role at your company for a long time, and how their responsibilities and role have changed? (I *love* this question)
- Do you have a dedicated designer? QA? Technical writer? Dev manager?
- How often do you have meetings? Are there any scheduled/standing meetings? Who talks to customers (if appropriate) and how?
- Has there been a situation where someone raised an ethical concern? If so, how was it handled? If not, have there really not been any?
- How are decisions made? Is architecture dictated top down? Are ideas from anyone welcomed? If so, in what scope/context?
- How are disagreements solved - both technical disagreements and other kinds? What happens when personalities clash?
- Can you tell me about a time when you've had to let someone go?
- Is there a written roadmap all developers can see? How far into the future does it extend? How closely is it followed?
- How is performance evaluated?

Quality of life

- How much vacation do people get? If there's "unlimited" vacation, how much vacation do people normally take?
- Is it possible to take sabbaticals or unpaid vacation?
- How many women work for you? What's your process for making sure you have diversity in other ways?
- How many hours do people work in an average week? In your busiest weeks?
- Is variability tolerated or is everyone expected to be on the same schedule?
- What time do people normally leave work?
- Would I need to be on call? How often?
- How often are there emergencies or times when people have to work extra hours?
- What is your turnover rate like? How many devs were hired last year and how many left?
- What's your retention rate of women over 1.5 years? Do you think you could have done anything differently to keep people who left?
- Do people work on the weekend?
- Do people check in when they're on vacation? How often?
- Is it possible to work from home, say, 1 or 2 days a week? Does anyone do this? (can be a nice option to have)
- Does this position require travel? How often?

As many of these as possible are "statistical" questions – a company may say that they "don't have hours", but if everyone leaves at 9pm that's not a good sign.

Community involvement

- Do you contribute to open source projects? Which projects? Which teams work on open source? Do you work mostly in the community or do you have a private fork?

- Do your employees speak at conferences about your work?

Career development

- Are employees encouraged to go speak at conferences?
- Do you cover travel to conferences? How many conferences a year do devs typically go to?
- Does your company support continuing education? (will they pay for employees to do a master's degree?)
- In what other ways do you support career development?

Culture

- How do you determine if someone is a poor fit for your company?
- How are your teams structured? What is the management structure like?
- How often do you pair? What's pairing like? How often do inexperienced people work directly with experienced people?
- What's the onboarding process like?
- Is there any sort of institutionalized way of dealing with plateauing or preventing burnout? (Expecting to hear about rotation of duties or location, sabbaticals.)
- Is it easy to move to other divisions or offices?
- How does internal communication work? This one is super important and I need to remember to ask it more.
- Are there catered suppers? (possibly bad)
- How many hours a week does senior management work? Do they put in 80-hour weeks?

Financials/business model/growth

- Are you profitable?
    - if not, how does this affect what you can do? What's your planned timeline for becoming profitable?
- How do you make money? (I often explain to my parents or non-technical friends companies' business models to test if they really make sense.)
- How much are you planning to hire in the next year?
- Are company financials, minus salaries, transparent throughout the company?

Things to look for in real life

- How is the office space physically organized?

Other

- Can you tell me about how the interview process is structured? How many interviews are there?
- How often do you offer above asking? Can I speak with someone who got such an offer?
- What do you wish you had known when you joined this company?
- Why did you choose to join this company?
- What's the single biggest issue or problem facing the team/department/company?" What's currently being done to address it?