# Job Interviews

Adrian Kosmaczewski

2021-01-22

If you work in this industry for a certain amount of time you are certainly bound to suffer, at some point or another, the delicious experience of the programmer job interview.

- How to prepare
- It is a two-way street
- Red lights
    - The interview is all about the company, little about you
    - The interviewer only pays attention to gaps in your CV
    - Few of the people interviewing you are technical folks
    - Watch out for simple lacks of decency
    - Stupid technical questions

Countless books have been written about the subject; everybody has their own opinion about it. And of course, dear reader, you are now going to hear my own: they all suck.

The programmer job interview is, at best, an exercise in futility. People who have never met before must decide, in a series of one to ten meetings of one hour each, if their future belongs together or not. Think speed dating minus the sex, minus the good food, minus the fun, minus everything, plus a contract and a salary.

Of course this process, largely driven by heuristics of different kinds, the least of which is rationality, can sometimes yield positive outcomes; yes, you can land a nice job through an interview. Inversely, if you are the employer, it might happen that you are lucky and you find the right employee.

But make no mistake, there is no scientific method in the job interview. There is guts, no data, no rationality. The problem with the job interview lies, once again, in hypocrisy. The moment things go south is when companies think they have a well rounded interview "process."

## How to prepare

The short answer is: do not. Just be yourself. The best job interviews are actually conversations between two like-minded people. You want smart people

in front of you, so just be yourself. I think that you do not prepare for a job interview; you just become, every day, a better software developer. You keep reading, you keep learning, and then every so often you go to an interview.

If you are yourself in an interview, you are making everyone a favour. They will see, in those short interactions, a faithful sample of your personality; your flaws and your capabilities. Being yourself sends a clear message: you do not settle for mediocrity or hypocrisy. You want like-behaved people in front of you; you know what you want, and you are actively looking for it.

Of course, it is to expect that the interviewers are also being themselves, but that is beyond your reach, so remember that the only thing you can do is pay attention, and, in the worst of cases, politely dismiss the interview and walk out.

## It is a two-way street

Most developers, particularly young ones, think that they should bend their wishes to any extent in order to get that job. Hint: you do not need to do that. A job interview is also a way for you to find out whether you want to work at that place, too, so you should be alert and watch out for any red lights.

## Red lights

Talking about signs that you should not sign up at a company, here is a couple few:

### The interview is all about the company, little about you

Some employers are so full of themselves (this is typically true with large, well known companies in Europe) that they will spend the whole time talking about their high standards, their achievements, and the kind of people that works for them. Pay attention to the wording: you can detect these folks because they say "people work *for* us" instead of "*with*" us. This subtle but important word choice reveals a lot about a company.

### The interviewer only pays attention to gaps in your CV

For some reason, some HR managers think that any period of time you do not mention in your CV is worth investigating. Maybe you took one year off, maybe you had family problems, maybe you just wanted to not work for a while. It is none of their business, and I actually think it is rather impolite (and, dare I say, the practice should be banned and outlawed.) Unfortunately in Europe personal questions are not illegal, so brace for impact. The interview should, at any time, be about what you have done, not what you have not done.

**Few of the people interviewing you are technical folks**

It is OK to have the occasional HR manager assessing your personality, of course; but more than 75% of all the people you are going to see in an interview should be developers, and not any developers, but potential co-workers from the team you would be assigned to. An interview is a rather short process, so the company should be optimizing the time spent in them.

**Watch out for simple lacks of decency**

Managers arriving late to the meeting room where the interview takes place; staff who do not even offer you a glass of water when you arrive to their offices; wrong directions to the location of the interview; comments about your appearance, nationality, or accent; managers ranting about their current staff or employer; inappropriate comments about race, gender, nationality or other non-technical factors.

As a rule of thumb, at the first occurrence of such events, one must wrap up the interview, greet the team and get out. If they behave this way in our first meeting, imagine what it will be when you sign the contract.

**Stupid technical questions**

Frankly, it is 2015 and companies are still asking how to write a linked list using C, on a whiteboard, without documentation, in one minute. This is nonsense. Whenever you are asked a question that could be answered by opening a book or searching in Stack Overflow, say so politely.

I tend to favor conversational topics, which I think make for a much better interview: design patterns, architecture, code quality issues, experiences in such and such frameworks, and so on. Conversational questions allow both parties to enter, guess what, in a conversation. And through that conversation, both parties can discover the other, the person on the other side of the table.

Most HR managers think that their job consists of finding people that allows them to cross checkboxes. That is wrong. In the words of Joel Spolsky, they should be looking for smart people who gets things done. And a sure way to find them is to actually talk to them, like human beings. Algorithms and data structures? Sure, there are plenty of literature about those; let's move on.