



Filtered

17 Insightful Questions

TO ASK A SOFTWARE ENGINEER CANDIDATE





Questions About Attitudes

1 How much time did you spend preparing for this interview?

If a candidate spent time preparing, it shows that they care about the role and other people's time. It also gives you a sense of their experience. More experienced candidates tend to prepare by doing research about the company or preparing their resume.

2 Tell me about a time that you failed.

The story is just as important as the way the story is told. You can quickly get a sense of the candidate's experience level, as well as how they feel about failure. Listen for where they place blame as well as what they learned.

3 Tell me about a time you did something that you were really proud of, but no one else knew about.

This question is good for getting to know the candidate's hobbies and interests that don't necessarily align with the job. Describing these types of projects will help you figure out the kind of work that keeps a candidate interested and challenged. Listen closely for how detail-oriented or how people-oriented the project was, and what motivated them to start the project.

4 When do you consider a project to be finished?

This is a question that uncovers experience and autonomy. Someone who is used to task-oriented or heavily-regulated work will likely have a definition of "finished". Someone who is business or product-oriented will more likely talk about how things are always changing.



5 What is required in order for a team to function successfully?

Some people have never thought about this, especially candidates that are typically focused on their own work. The more that the candidate's answer relates to them, and the less it relates to an entire team or organization, the more inwardly-focused they may be.

6 Tell me about a time where a coworker frustrated you. How did you handle it?

This question is very similar to the question about a successful team. Look for clues of what the candidate considers to be a conflict. Was it a disagreement in personalities? Was it a miscommunication? You can identify some soft skill gaps by asking this question.

7 Have you ever disagreed with your boss or manager? What happened?

While you're not looking for someone who is hard to work with, questions about disagreement with a manager are a good way to understand how the candidate navigates company politics. Do they always back down? Do they always get defensive? You're looking for empathy and the ability to resolve conflict through compromise.



FILTERED PRO TIP

Overcoming Bias

Human beings are more likely to “like” the people who are closest in personality and attitude to themselves. This is a well-known bias. Try to overcome this by comparing your candidate's statements to an objective written rubric. Finding a way to automate this rubric is the best way to stay unbiased.



Questions About Experience

8 Have you ever had to be on-call? How did it work?

Being on-call usually comes with a decent amount of responsibility. If the candidate answers “yes” then they have likely already demonstrated some sort of autonomy and trust of their team (if applicable). Some types of engineers don’t ever have to be on-call, so this question may not apply to them.

9 Have you ever had a bug in your code that showed up in production? What happened? What did you learn?

Bugs are bad, obviously. What you should look for is not necessarily the type of bug, but the attitude that the engineer has about it. Less experienced engineers tend to distance or blame themselves from the “mistake” of a bug, and more experienced engineers think of bugs in terms of the severity, and scope of the product. Listen for stories about preventing future bugs, which shows that they understand how to re-prioritize their work after an incident.

10 Have you ever done a post mortem? If so, tell me about one.

Post-mortems are a great way to reflect on things that didn’t go as planned. Listen for the role that this candidate played in this post-mortem. Did they contribute to the recap or the solution? How did they feel about being accountable? Less experienced candidates may not have had the chance to use post-mortems in their work, which is okay.



11 How would you describe the software lifecycle at your last position? What did you like about it? What did you wish you could change?

This question is intentionally high-level. The candidate will answer by demonstrating their knowledge of what happens to software before and after it's built. Some candidates who are focused on their own work may not know the answer to this question, while others may enjoy talking about all of the moving parts of the business and how they relate. This question can help you fit a candidate into the right role.

12 Have you mentored or taught others who are more junior than you? What was it like?

Teaching experience is always a plus because it benefits the candidate as much as it benefits the student. Even just one experience with teaching can demonstrate that a candidate is approachable, autonomous, trusted, and able to communicate complex topics in a way that someone less experienced can understand (you know, like to a salesperson or product manager).



FILTERED PRO TIP

Spotting Dishonesty

Sometimes candidates aren't totally honest about their experience – let's be honest... it's quite often. Using questions that force a candidate to tell a story and be specific can help you get a better read on the true level of experience. Time is not the only factor when it comes to experience: challenging problems, helpful peers, and relevant side projects all pointing to deeper knowledge of technology.



Questions About Technical Skills

13 What does a good testing process look like?

Engineers that build solid systems will always have an opinion about testing. Writing bug-free code is not magic, it's just about being prepared. Ask this question and let the engineer ramble a bit about what they like and what they don't. Listen for stories about situations that required better testing and how they handled it.

14 What are the pros and cons of monolithic vs. microservice architectures?

Just like the question about the software lifecycle, this question is also high-level. Engineers with more experience are able to describe these popular architectures in a way that relates to the business as a whole, not just immediately related to their code. The younger the engineer is, the less likely that they have worked with both types of architectures.

15 What do you like about ____ and what would you change about it?

This question is *really good* for catching BS'ers. Experienced engineers will be more likely to talk about how the technology relates to other similar technologies in terms of implementation or performance, as well as nitpick details based on experienced that they have had. Less experienced (or those with no experience) lean heavily on the surface-level details like syntax or popular libraries without personal accounts to back them up.



16 Walk me through how you learn new skills.

The skills that the candidate mentions are not the important part of this question. Instead, you'll want to listen for the frequency of learning, any attempts to use the skills, and the depth to which they attempt to learn. Strong engineering candidates are usually “learners” who can go deep into new concepts, which protects them from becoming limited to only legacy technologies in the future.

17 I've heard that you can write a working ___ in 2 lines of ___ code. Is that true?

This is kind of a trick question. By asking this question, you are being intentionally naive in order to get a response from your candidate. Whatever happens, you should listen for the explanation of why you are right or wrong. Inexperienced candidates are more likely to take this question as a quiz that has a right or wrong answer without being as concerned about why. More experienced candidates or younger candidates with potential are more likely to analyze in order to get to the correct answer.



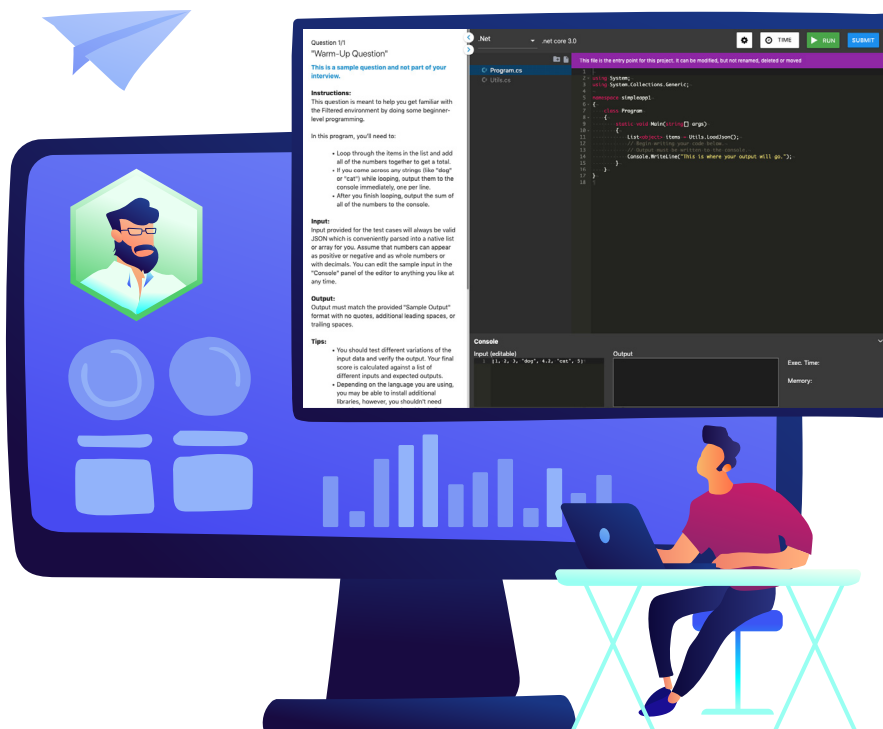
FILTERED PRO TIP

Evaluation Techniques

It's basically impossible to evaluate a candidate's technical skill over the phone, or even in front of a whiteboard. Many modern companies are giving candidates take-home code tests (which need to be graded), or have them do paired programming during the interview (which takes at least an hour). The best solution to this problem is to automate code testing with software—this gives you automatic scoring and analysis.

Test your candidates before they meet you.

Filtered enables your business to overcome today's tech hiring challenges and assess a candidate's true fit for the job, team, and company through capabilities-based hiring. Our approach uncovers previously hidden applicant insights and measures their true match, not only to the job but to your business goals.



CHECK US OUT AT [FILTERED.AI](https://filtered.ai)



Filtered