

V. Behavioral Questions | Behavioral Preparation

Behavioral questions are asked for a variety of reasons. They can be asked to get to know your personality, to understand your resume more deeply, or just to ease you into an interview. Either way, these questions are important and can be prepared for.

How to Prepare

Behavioral questions are usually of the form “Tell me about a time when you...,” and may require an example from a specific project or position. I recommend filling in the following “preparation grid” as shown below:

Common Questions	Project 1	Project 2	Project 3	Project 4
Most Challenging				
What You Learned				
Most Interesting				
Hardest Bug				
Enjoyed Most				
Conflicts with Teammates				

Along the top, as columns, you should list all the major aspects of your resume, including each project, job, or activity. Along the side, as rows, you should list the common questions: what you enjoyed most, what you enjoyed least, what you considered most challenging, what you learned, what the hardest bug was, and so on. In each cell, put the corresponding story.

In your interview, when you’re asked about a project, you’ll be able to come up with an appropriate story effortlessly. Study this grid before your interview.

I recommend reducing each story to just a couple of keywords that you can write in each cell. This will make the grid easier to study and remember.

If you’re doing a phone interview, you should have this grid out in front of you. When each story has just a couple of keywords to trigger your memory, it will be much easier to give a fluid response than if you’re trying to re-read a paragraph.

It may also be useful to extend this grid to “softer” questions, such as conflicts on a team, failures, or times you had to persuade someone. Questions like these are very common outside of strictly software engineer roles, such as dev lead, PM or even testing role. If you are applying for one of these positions, I would recommend making a second grid covering these softer areas.

When answering these questions, you’re not just trying to find a story that matches their question. You’re telling them about yourself. Think deeply about what each story communicates about you.

V. Behavioral Questions | Behavioral Preparation

What are your weaknesses?

When asked about your weaknesses, give a real weakness! Answers like “My greatest weakness is that I work too hard” tell your interviewer that you’re arrogant and/or won’t admit to your faults. No one wants to work with someone like that. A better answer conveys a real, legitimate weakness but emphasizes how you work to overcome it. For example: “Sometimes, I don’t have a very good attention to detail. While that’s good because it lets me execute quickly, it also means that I sometimes make careless mistakes. Because of that, I make sure to always have someone else double check my work.”

What was the most challenging part of that project?

When asked what the most challenging part was, don’t say “I had to learn a lot of new languages and technologies.” This is the “cop out” answer when you don’t know what else to say. It tells the interviewer that nothing was really very hard.

What questions should you ask the interviewer?

Most interviewers will give you a chance to ask them questions. The quality of your questions will be a factor, whether subconsciously or consciously, in their decisions.

Some questions may come to you during the interview, but you can—and should—prepare questions in advance. Doing research on the company or team may help you with preparing questions.

Questions can be divided into three different categories.

Genuine Questions

These are the questions you actually want to know the answers to. Here are a few ideas of questions that are valuable to many candidates:

1. “How much of your day do you spend coding?”
2. “How many meetings do you have every week?”
3. “What is the ratio of testers to developers to program managers? What is the interaction like? How does project planning happen on the team?”

These questions will give you a good feel for what the day-to-day life is like at the company.

Insightful Questions

These questions are designed to demonstrate your deep knowledge of programming or technologies, and they also demonstrate your passion for the company or product.

1. “I noticed that you use technology X. How do you handle problem Y?”
2. “Why did the product choose to use the X protocol over the Y protocol? I know it has

V. Behavioral Questions | Handling Behavioral Questions

benefits like A, B, C, but many companies choose not to use it because of issue D.”

Asking such questions will typically require advance research about the company.

Passion Questions

These questions are designed to demonstrate your passion for technology. They show that you’re interested in learning and will be a strong contributor to the company.

1. “I’m very interested in scalability. Did you come in with a background in this, or what opportunities are there to learn about it?”
2. “I’m not familiar with technology X, but it sounds like a very interesting solution. Could you tell me a bit more about how it works?”

V. Behavioral Questions | Handling Behavioral Questions

As stated earlier, interviews usually start and end with “chit chat” or “soft skills.” This is a time for your interviewer to ask questions about your resume or general questions, and it’s also a time for you ask your interviewer questions about the company. This part of the interview is targeted at getting to know you, as well as relaxing you.

Remember the following advice when responding to questions.

Be Specific, Not Arrogant

Arrogance is a red flag, but you still want to make yourself sound impressive. So how do you make yourself sound good without being arrogant? By being specific!

Specificity means giving just the facts and letting the interviewer derive an interpretation. Consider an example:

- Candidate #1: “I basically did all the hard work for the team.”
- Candidate #2: “I implemented the file system, which was considered one of the most challenging components because ...”

Candidate #2 not only sounds more impressive, but she also appears less arrogant.

Limit Details

When a candidate blabbers on about a problem, it’s hard for an interviewer who isn’t well versed in the subject or project to understand it. Stay light on details and just state the key points. That is, consider something like this: “By examining the most common user behavior and applying the Rabin-Karp algorithm, I designed a new algorithm to reduce search from $O(n)$ to $O(\log n)$ in 90% of cases. I can go into more details if you’d like.” This demonstrates the key points while letting your interviewer ask for more details if he wants to.

Give Structured Answers

There are two common ways to think about structuring responses to a behavioral question: nugget first and S.A.R.. These techniques can be used separately or in together.

Nugget First

Nugget First means starting your response with a “nugget” that succinctly describes what your response will be about.

For example:

- Interviewer: “Tell me about a time you had to persuade a group of people to make a big change.”
- Candidate: “Sure, let me tell you about the time when I convinced my school to let undergraduates teach their own courses. Initially, my school had a rule where...”

V. Behavioral Questions | Handling Behavioral Questions

This technique grabs your interviewer's attention and makes it very clear what your story will be about. If you have a tendency to ramble, it also helps you be more focused in your communication, since you've made it very clear to yourself what the gist of your response is.

S.A.R. (Situation, Action, Result)

The S.A.R. approach means that you start off outlining the situation, then explaining the actions you took, and lastly, describing the result.

Example: "Tell me about a challenging interaction with a teammate."

- **Situation:** On my operating systems project, I was assigned to work with three other people. While two were great, the third team member didn't contribute much. He stayed quiet during meetings, rarely chipped in during email discussions, and struggled to complete his components.
- **Action:** One day after class, I pulled him aside to speak about the course and then moved the discussion into talking about the project. I asked him open-ended questions about how he felt it was going and which components he was most excited about tackling. He suggested all the easiest components, and yet offered to do the write-up. I realized then that he wasn't lazy—he was actually just really confused about the project and lacked confidence. I worked with him after that to break down the components into smaller pieces, and I made sure to compliment him a lot on his work to boost his confidence.
- **Result:** He was still the weakest member of the team, but he got a lot better. He was able to finish all his work on time, and he contributed more in discussions. We were happy to work with him on a future project.

The situation and the result should be very succinct. Your interviewer generally does not need many details to understand what happened and, in fact, may be confused by them..

By using the S.A.R. model with clear situations, actions and results, the interviewer will be able to easily identify how you made an impact and why it mattered.