Sample Prior Experience Questions

In this lesson, we'll look in-depth at some prior experience questions.

We'll cover the following

- Sample questions
- Sample answers

Sample questions

Prior experience questions come in all shapes and sizes. We have crafted some example ones (below) for you to try your hand at. Go ahead and answer them aloud!

- What was the last project you led, and what was its outcome?
- Give me an example of a time that you felt you went above and beyond at work.
- Describe a time when your work was criticized, how you responded, and what happened as a result.
- Have you ever been on a team where someone was not doing their part? How did you handle it?
- Tell me about a time when you had to give someone difficult feedback. How did you handle it?
- How do you handle working with people who annoy you?
- What was the most difficult period in your life, and how did you deal with it?
- Give me an example of a time you did something wrong. How did you handle it?
- Tell me about a time when you had to deal with conflict on the job.
- What assignment was too difficult for you, and how did you resolve the issue?
- Give me an example of a difficult decision you had to make. How did you come to that decision?
- Give an example of an idea you implemented.
- Describe a mentor who has impacted you in a positive way.
- Was there a person in your career who really made a difference? In what way?
- Tell me about your proudest

achievement.

Sample answers

Q: “When things go wrong, how do you deal with it? Give me an example of a situation you dealt with in a previous role.”

Hide

The classic scenario is the demo that stops working right before your big presentation. This happened to me when I was at ABC Company(**Situation**). I was leading the development of a new mobile experience, and we were scheduled to meet with senior leaders to demo our V0 prototype and get approval for V1 funding. We were having issues with it in the hour before the meeting. I was responsible for presenting the business case and the designer was going to run the demo(**Tasks**). We needed time to get to the meeting, so we left the engineers working on it. I met with one senior leader ahead of time to make sure we had a champion in the room. I knew she expected to see the same demo. I felt a lot of pressure. Ten minutes before the meeting, it still wasn’t working, so I suggested to the designer that we present the storyboard mockup instead(**Actions**). We had an electronic version that was full of photos and illustrations. I added the customer persona and journey maps to my presentation. The designer and I both focused on creating a narrative centered on our customer and her pain points. I whispered in the ear of our champion when she arrived and told her we had modified our presentation approach, but that the facts were all the same(**Results**). Our presentation became “asking for advice” about our V0 prototype instead of “asking for approval.” We asked stakeholders for suggestions and they gave us great ideas about go-to-market, monetization, and positioning. In the end, they approved more funding for V1 than our original request. I was reminded of the value of a rough prototype and asking for advice.

Be ready for these possible follw-up questions:

- What would have been different if the prototype had worked perfectly?
- What would you do to prevent this from happening in the future?

Hide

I was given a stretch assignment to lead the turnaround of a troubled project. The team had launched an ecommerce product that was doing great in terms of sales, but taking too many resources to support. Instead of operationalizing a successful project, the team was frustrated and burned out(**Situation**). I spent the first week interviewing team members one-on-one. During the second week, I decided to attend a weekly Thursday night support call. These calls were scheduled to update the site with new products, offers, and pricing. This was supposed to be a routine maintenance activity, but it was notorious for failures that led to all-nighters for the team. I was listening in to learn about the processes(**Tasks**). The database job completed as expected with a 'job ran successfully' message and we thought it might be a short call. Then, quality-checking revealed missing updates. Re-running the job gave the same result. The software we were using was a third party service. They were on the call, but unsure how to proceed. At this point, I recognized what the team had been telling me. When a problem occurred, the SRE's and DBA's were expected to stay on the call until the problem was resolved. This led to wasted time and lots of frustration. Instead, I decided to take a different action(**Actions**). I confirmed the escalation process with the third party team. It was clear that they needed time to troubleshoot. I told everyone else to go to bed and the Help Desk would contact the engineer on call if their team was needed. I gave the third party engineers instructions to call me directly with updates and confirmed that I would coordinate getting resources on our side when needed.

After 6 hours, the problem was fixed by the third party. I contacted an SRE and DBA on our side to re-run the job and validate(**Results**). The more important result was that I established my role as a leader who would listen to the team, hold vendors accountable, make decisions, and facilitate problem-solving without wasting their time. I made a lot of fans that night, which was important because I needed every one of them to turn the project around.

Be ready for these possible follow up questions:

- How long did it take to complete the turnaround?

- What were the root causes of the project failure?

Hide

I was working as an App Developer for XYZ Company when a Zero-Day Privilege Elevation vulnerability was discovered. Microsoft issued a patch and we had 6000 Windows servers running in 4 different data centers that all needed to be patched as quickly and efficiently as possible (**Situation**). It wasn’t my job to patch servers, but the VP asked all of us for ideas to get the work done faster (**Tasks**). I suggested that several of us could help do validation if the procedure was documented. Others could help with tracking the servers coming in and out of rotation and updating tickets. This freed up the Systems Engineers to do the work that only they could do (**Actions**). I had never done some of those tasks before, so I had to follow procedures exactly and ask for help if I wasn’t sure about something. I was willing to take directions and learn. I took good notes and helped make the documentation easier to follow (**Results**). I gained a new appreciation for the Systems and Security Engineers. We worked long hours, but it felt good to pull together. It was also good to know that we could do that again if a serious security issue came up.

Be ready for possible follow up questions:

- How did you manage separation of duties?
- How long did it take overall?

Hide

One of my leads sent me a text one night asking me to join a Support call. My team managed the business applications for Acme. It was end-of-month and payroll processing time. Jobs were failing and there was a sense of panic on the call (**Situation**). I was responsible for the execution of the payroll run within a specific time window (**Tasks**). I announced myself on the call and asked for a situation report. This is where my military training kicked in. My asked for a situation report. This is where my military training kicked in. My team is used to me being a laid-back manager, but I find that people calm

down when you ask methodical questions and give clear and specific directions. The team was not able to give a clear description of “last known good,” so I started dispatching people to get details and report back. Instead of just verbally reporting, I asked everyone to update a shared doc so that we could all see the knowns and unknowns as they came in (**Actions**). My team had guessed prematurely that the problem was in the most recent code release, but reviewing all recent changes showed us that there had been a change for scheduled backups that was creating db contention and jobs were timing out (**Results**). It didn’t take long to solve the problem after that. Being calm under pressure helps other people be calm. I can be easy-going most of the time, but I am still very direct when necessary. When there’s a problem, people like knowing someone is in charge. To avoid panic the next time, I facilitated a post-mortem to document troubleshooting steps and came up with a process to designate a leader during a live issue.

Be ready for these possible follow up questions:

- What’s your onboarding process for new resources?
- How else could you improve processes to prevent a recurrence?

Try your hand at perfecting your prior experience responses in the next lesson!