

# Interview questions and how to answer them with PAR

A practical guide for technology professionals interviewing at international companies: how to open with a strong Hook, answer behavioral questions with PAR, and avoid the language habits that quietly cost offers.

12-minute read · Free · By Mark Parfenov

## IN THIS GUIDE

Why good candidates lose offers

How an interview loop works

“Tell me about yourself”: the Hook

The PAR method

Your PAR story bank

Motivation questions

Strengths and weaknesses

Salary and logistics

Language that costs offers

Follow-ups and pressure tests

Your questions for them

Practice and final checklist

[Start here](#)

## Why good candidates lose offers

Most candidates prepare for interviews by memorizing their resume. That's a mistake. Hiring managers don't care about a chronological list of your past jobs. They care about the problems you know how to solve, and whether you can explain how you solved them.

If you're interviewing in English as a second or third language, you're doing two jobs at once: answering the question, and translating yourself (your ideas, your value, your judgment) into a language and a cultural style the interviewer recognizes. Strong engineers, designers and product people lose offers not because they lack skill, but because that second job isn't working.

The fix is structure. Walk in with **one strong Hook** and **three to five PAR stories** mapped to the job description, and you control the flow of the conversation instead of reacting to it. This guide shows you how to build both.

01

## How an interview loop works

Most international tech companies run a loop of four or five stages, often within two to four weeks. Each stage tests something different, so prepare for each one separately.

01

**Recruiter screen**

15–45 minutes. Fit, motivation, experience at a high level, salary and logistics.

02

**Hiring manager**

Your experience in depth: ownership, impact and how you work with others.

03

**Technical or case**

Coding, system design, a product or design case. How you think, not just the answer.

04

**Behavioral**

Conflict, failure, feedback and leadership, told as specific stories.

05

**Final round**

A holistic check with a senior leader: judgment, goals and team fit.

**TIP**

Keep the job description open while you prepare, and during phone screens. Using the exact language of the posting creates an immediate connection between what they need and what you say.

02

**“Tell me about yourself”: the Hook**

When an interviewer says “Tell me about yourself”, they aren’t asking for your life story. They’re handing you a blank canvas to frame the entire conversation. Most people waste it: where they studied, a list of technologies, every job in order.

A strong Hook does three things in under 45 seconds:

01

**The core problem**

Define yourself by the problem you solve, not the tools you use.

02

**The proof**

One sentence with your biggest, most relevant win.

03

**The bridge**

Connect that proof to this team: name their product, their challenge, and your objective.

**THE FORMULA**

“I solve **[core problem]**. I proved this at **[past company]** by **[the proof]**. I’m here because I want to bring that experience to **[their product or challenge]**.”

**WEAK**

“I’ve been a software engineer for six years. I started at a startup doing Python, then moved to another company doing Java. Now I’m looking for a senior role where I can grow.”

**STRONG**

“I specialize in building highly reliable distributed systems. At my last company, I designed an event-driven architecture that processed millions of messages a day with zero data loss. I know your team is moving to a Kafka-based backend, and I want to bring my experience with exactly those scalability challenges to help you ship it faster.”

After the Hook, you can extend to a 60–90 second answer by adding one or two highlights from your recent roles, each one a mini-PAR, and ending on why this role is the logical next step.

**More examples**

These examples are illustrative. Build yours from your own real results.

## Product Manager

I help B2B products grow revenue from the customers they already have. At my current company I own pricing and packaging for our analytics platform, and last year I redesigned our upgrade path, which lifted expansion revenue by 30% in two quarters. I saw that you're launching an enterprise tier this year, and that's exactly the kind of problem I want to work on next.

## UX/UI Designer

I design complex workflows so that people can finish them without calling support. At a logistics SaaS company, I rebuilt the shipment-booking flow from eleven steps to four, and support tickets about booking fell by more than half. Your platform is expanding to small businesses who won't have a dedicated operations team, and making that first booking effortless is where I can contribute most.

## Data Analyst

I turn messy product data into decisions that teams actually act on. At my current company, I built the retention dashboard that leadership now uses every week, and my churn analysis led to a change in onboarding that reduced early cancellations. You're scaling into new markets, which means a lot of new questions and not much clean data yet, and that's the kind of environment I work best in.

03

## The PAR method

For most tech roles I recommend PAR: **Problem, Action, Result**. When a story only makes sense with some background, use SAR instead: **Situation, Action, Result**, with the situation kept to two sentences. Longer frameworks tend to make people spend too long on the setup and not enough on what they actually did. PAR forces you to get to the substance quickly, which matters even more when you're speaking in a second language and every extra sentence is a chance to lose the thread.

01

### Problem · ~20%

Not just what you were building: the constraint that made it hard: time, budget, performance, bad data, competing stakeholders.

02

### Action · ~60%

What *you* decided and did, and why. The trade-offs you considered, what you chose not to do, how you handled what went wrong.

03

### Result · ~20%

What changed for users or the business. Numbers are best; clear descriptive impact also works.

SAY "I", NOT "WE"

"We worked hard and used Agile" tells the interviewer nothing about you. Describe your decisions, even inside a team effort.

## A PAR story, start to finish

**Software Engineer: “Tell me about a difficult technical problem you solved.”**

**PROBLEM**

We needed a caching layer on an edge device, but we only had 50 megabytes of memory available, and the device kept crashing under load.

**ACTION**

I chose an LRU cache instead of a simple array, because it kept lookups constant-time. To handle the memory limit, I wrote a custom eviction policy that cleared stale data before the operating system could kill the process.

**RESULT**

We cut p99 latency by 40% and eliminated out-of-memory crashes in production. Getting there took some interesting low-level memory work.

### How to deliver it

- **Keep it under two minutes.** A PAR story is a concentrated shot of proof. If you reach two minutes, stop.
- **Signpost the start.** “The best example of that is...” or “That reminds me of a project where...” tells the listener a structured answer is coming.
- **Tee up the follow-up.** End with a detail that makes the interviewer curious, like “some interesting low-level memory work” above. They’ll ask about it, and now you’re in a conversation instead of reciting an answer.

04

## Your PAR story bank

Behavioral questions (“Tell me about a time when...”) are where many offers are decided, especially with hiring managers. Interviewers want specific examples, not general statements about how you usually behave. The way to be ready is a story bank: five stories prepared in advance, each flexible enough to answer several questions.

- ☐ **Conflict:** a disagreement with a manager, colleague or stakeholder
- ☐ **Failure:** a mistake you owned and what you changed afterwards
- ☐ **Feedback:** difficult feedback you received and acted on
- ☐ **Leadership:** leading people or a project, with or without authority
- ☐ **Delivery under pressure:** a tight deadline, limited resources, a hard problem

The same five stories cover most of the questions you’ll hear:

- Tell me about a time you disagreed with your manager.
- Describe a situation where you had to convince someone to support your idea.
- Tell me about a time you failed, or made a significant mistake.
- Give me an example of difficult feedback you received.
- Describe a project you led without formal authority.
- Tell me about a time you delivered under a very tight deadline.
- Tell me about a recent project you’re proud of.

### Example PAR stories

Illustrative examples: replace every detail with your own.

### Conflict · Backend Engineer: “Tell me about a time you disagreed with a senior colleague.”

#### PROBLEM

Our lead architect wanted to rewrite our billing service from scratch. I thought a full rewrite would freeze new features for at least two quarters, and billing was already blocking two product launches.

#### ACTION

Instead of arguing in the meeting, I asked for a week to test an alternative. I built a small proof of concept that moved one billing flow to a new module behind the existing interface, measured the effort, and presented both plans side by side with timelines and risks. I was clear that I'd fully support whichever plan we chose.

#### RESULT

We went with the incremental migration. Both launches shipped on time, and the old service was fully replaced eight months later. The architect and I now review major design decisions together.

### Failure · DevOps Engineer: “Tell me about a mistake you made.”

#### PROBLEM

I pushed a configuration change on a Friday afternoon without a staged rollout, and it took down checkout for twenty minutes.

#### ACTION

I rolled it back, told my manager and the on-call channel exactly what I'd done, and led the incident review the following Monday. I didn't stop at “be more careful”: I proposed and built a canary step in our deployment pipeline so configuration changes reach 5% of traffic first.

#### RESULT

We haven't had a configuration-related outage since, and the canary step is now standard for every team. It taught me that owning a mistake quickly is what makes people trust you with the next big change.

### Feedback · Product Manager: “Give me an example of difficult feedback you received.”

#### PROBLEM

My director told me my updates to leadership were too long and that people stopped reading before they reached the decision I needed.

#### ACTION

I restructured every update to lead with the bottom line: the decision or risk in the first sentence, then three supporting points, then details only for those who wanted them. I asked my director to review my next three updates and tell me honestly whether they worked.

#### RESULT

Decisions that used to take a week of back-and-forth started getting answered the same day, and my format became the template for the other PMs in the group.

### Leadership · Product Designer: “Describe a project you led without formal authority.”

#### PROBLEM

Three product teams were building similar interface components separately, so the product looked inconsistent and every team was repeating the same work.

#### ACTION

I audited the duplicated components, calculated how many hours each team was losing, and brought the numbers to the three engineering leads. I proposed a shared component library and started with the five components everyone used, so the value was visible within a month.

#### RESULT

All three teams adopted the library within a quarter, new screens took noticeably less time to build, and I was asked to lead the design system officially.

#### WANT FEEDBACK ON YOUR STORIES?

Book a free consultation. Bring one of your PAR stories and I'll tell you what an interviewer will hear, and how to make it stronger.

[Book a free call](#)

05

## Motivation questions

Recruiters want to know you're applying to them on purpose, not sending the same resume everywhere. Expect some version of:

- **"What do you know about our company?":** show you've researched the product, the market and recent news.
- **"Why this position?":** connect your goals to the specific requirements in the job description.
- **"Why are you leaving your current role?":** give a positive, forward-looking reason: growth, bigger challenges, more impact.

#### RED FLAGS

- Mentioning perks, salary or work-life balance as your main reason
- Generic praise like "great culture" or "market leader"
- Complaining about your current manager or company
- Saying you want "less stress" or "something easier"

#### GENERIC

"You're a great company with a great culture, and I want to grow."

#### SPECIFIC

"You're moving from a single product to a platform, and I've spent the last two years building the internal APIs that make that shift possible. That's the problem I want to work on next."

#### BACKWARD-LOOKING

"My manager doesn't listen and there's no structure."

#### FORWARD-LOOKING

"I've learned a lot there, and I'm now ready for a role with more ownership of product decisions, which is exactly what this position offers."

06

## Strengths and weaknesses

For strengths, choose two that the job description clearly needs and back each one with a quick, measurable example. A strength without proof sounds like a slogan.

For weaknesses, interviewers are testing self-awareness, not looking for a confession. Use a simple structure:

01

#### A real weakness

Something genuine that doesn't disqualify you from the core of the role.

02

#### What you do about it

A specific habit or system, not "I'm working on it".

03

#### Evidence of progress

A recent example that shows it's improving.

### Example: Frontend Engineer

Earlier in my career I'd stay quiet in design reviews when I disagreed, and raise concerns later, which meant rework. Now I write down my main concern before each review and make sure I say it in the meeting, framed as a question. In our last planning cycle, that's how we caught an accessibility problem before any code was written.

Avoid "I'm a perfectionist" or "I work too hard". Interviewers have heard them hundreds of times, and they signal that you haven't really thought about the question.

07

## Salary and logistics

These "knockout" questions usually come in the recruiter screen. A mismatch can end the process immediately, so know your answers before the call.

- **Salary expectations:** give a grounded range and be ready to justify it with market data and the impact you bring.
- **Work authorization:** be clear about whether you need sponsorship in that country.
- **Notice period and start date:** know it exactly; fast-moving companies notice flexibility.
- **Tools and technologies:** be honest about your level with the non-negotiables in the posting.

08

## Language that costs offers

For non-native speakers, the biggest problems are rarely grammar. They're habits that make a strong candidate sound less confident, less senior or less clear than they are.

| LESS EFFECTIVE                                                 | MORE EFFECTIVE                                                |
|----------------------------------------------------------------|---------------------------------------------------------------|
| "I think maybe we could potentially try a different approach." | <b>"I recommended a different approach."</b>                  |
| "We did a migration and it went well."                         | <b>"I led the migration and cut deployment time by half."</b> |
| "Sorry, maybe I didn't explain it correctly."                  | <b>"Let me put that another way."</b>                         |
| Two minutes of background before the point                     | <b>The conclusion first, then the context</b>                 |
| "I'm not sure, but I think it was difficult."                  | <b>"The main constraint was..."</b>                           |

Culture matters too. What sounds confident in one market can sound arrogant or vague in another: German interviews tend to reward formality and precision, Dutch interviews directness, US interviews visible enthusiasm, and UK interviews understatement. Research the norms of the market you're targeting and adjust your tone, without pretending to be someone you're not.

09

## Follow-ups and pressure tests

Good interviewers interrupt and dig deeper: "Why didn't you just do the simpler thing?" or "That sounds like your manager's decision. What did *you* do?" This isn't hostility; it's how they test whether your story is real.

- Stay calm, and defend your decisions with reasoning and data.
- If you need a moment, say so: "That's a good question, let me think for a second."
- In technical rounds, think out loud and ask clarifying questions before you start.
- Offer a working, pragmatic solution first, then discuss how you'd scale or improve it.

10

## Your questions for them

"Do you have any questions for me?" is part of the interview. Saying "no" is a missed opportunity, and at some companies, a red flag. Ask questions that show you're already thinking about the work:

- What is the biggest bottleneck the team is facing right now?
- How will success in this role be measured in the first six months?
- What does a typical week look like for the team?
- What's the most common reason people don't succeed in this role?
- What are the next steps and the timeline for the process?

11

## Practice and final checklist

Knowing what to say is only half the work. The other half is saying it clearly under pressure, and that only comes from practice out loud.

- **Record yourself.** Answer a question on your phone and listen back. You'll hear exactly where you hesitate, over-explain or lose the thread.
- **Do it twice.** After you listen, answer the same question again with the corrections. The second attempt is where the improvement actually happens.
- **Run a full mock interview.** Practicing individual answers isn't the same as holding structure for 45 minutes with someone asking follow-ups.

### Before your interview, check that you can:

- ☐ Deliver your Hook in under 45 seconds, naming their product or challenge
- ☐ Tell five PAR stories (conflict, failure, feedback, leadership, delivery), each in under two minutes
- ☐ Explain why this company and this role without mentioning perks
- ☐ State your salary range, work authorization and notice period without hesitating
- ☐ Say "I" when describing your contribution
- ☐ Ask three thoughtful questions at the end

Under pressure, you won't rise to the level of your expectations. You'll fall to the level of your preparation.

PRACTICE WITH FEEDBACK

## Build your Hook and PAR stories with me

I'll help you shape your stories, run full mock interviews with you and work on your language and delivery, before the interview that matters.

[Book a free call](#)