

UX Designer interview questions

careertips.com.au: common questions and what they're really testing

What to expect

UX Designer interviews usually mix a portfolio review with behavioural questions and at least one design exercise, since employers want to see both how you think and what you've actually shipped. Expect the panel to include a hiring manager plus someone from product or engineering, given how much of the role is collaborative.

- **Portfolio walkthrough:** You talk through a real project end to end, including the messy parts, not just the polished final screens.
- **Behavioural:** Past-experience questions about handling feedback, disagreement or ambiguous requirements.
- **Scenario / design exercise:** A short brief, sometimes done live or as a take-home, testing how you approach a problem under real constraints.
- **Process and tools:** Questions about your research methods, prioritisation approach and how you document and hand off designs.
- **Collaboration and stakeholder:** How you work with product managers, developers and business stakeholders who may not share your priorities.

Most processes start with a recruiter or hiring manager screen, followed by a portfolio presentation to the design team, then a scenario exercise or take-home task reviewed in a follow-up session, and often a final panel round with product and engineering stakeholders.

1. Walk us through one project in your portfolio, from the initial problem to the final design.

Why they ask

They want to see your process, not just the finished screens, and how you handled ambiguity along the way.

How to structure your answer

Present it chronologically: the problem or business need, how you researched it, the options you explored, why you landed on the final design, and what you'd do differently now.

Example answer

"I'll walk through a redesign of a booking flow that had a high drop-off rate at the final step. I started with a handful of user interviews and a review of session recordings to see where people were getting stuck. That pointed to a confusing pricing summary, so I sketched a few layout options, built a clickable prototype in Figma, and tested it with the same group of users. The version that grouped costs by category tested best, so I refined that into the final design and documented the pattern for the team's design system. Looking back, I'd have brought a developer into the testing sessions earlier, since a couple of the layout choices needed technical trade-offs I only found out about after the fact."

2. Tell me about a time a product manager or developer pushed back on your design and you had to change your approach.

Why they ask

UX work involves constant negotiation with people who have different priorities, and they want to know you can adapt without just caving or digging in.

How to structure your answer

STAR: describe the situation, the specific disagreement, the action you took to resolve it, and the result, including what you learned about working with that stakeholder.

Example answer

"I'd designed a multi-step onboarding flow based on user testing that showed people preferred a guided walkthrough. A developer pushed back because the timeline didn't allow for that much custom build work. Instead of insisting on the original design, I went back to the research and identified which parts of the walkthrough were actually driving the improvement, then proposed a simpler version that kept those elements but reused existing components. We shipped the simplified version on time, and I ran a smaller usability check afterwards to confirm it still solved the core problem."

3. You've got two weeks to improve a feature's usability, but you can't get direct access to users. How would you approach it?

Why they ask

Real projects often come with time and access constraints, and they want to see how you make good decisions with incomplete information rather than stalling.

How to structure your answer

Talk through your judgement under pressure: what you'd rely on instead of live user access, how you'd sequence the work, and how you'd flag the limitations of your approach to stakeholders.

Example answer

"I'd start with what's already available: support tickets, analytics on where people drop off, and any past research on that feature. I'd use that to form a hypothesis about the main friction points, then design two or three options and run a quick unmoderated test through a tool like UserTesting rather than trying to schedule full moderated sessions. I'd be upfront with the team that this is a lighter-weight validation than I'd normally want, and flag it as something to revisit with proper research once the deadline pressure eases."

4. How do you decide which usability findings to act on first when you've got a long list of issues?

Why they ask

They want to know you can prioritise sensibly rather than trying to fix everything at once or chasing the loudest complaint.

How to structure your answer

Describe your prioritisation framework step by step: how you weigh severity, frequency and effort, and how you communicate the resulting order to stakeholders.

Example answer

"I usually sort findings by how often users hit the issue and how badly it blocks them from completing their task, rather than just how easy the fix looks. A rare but task-blocking issue gets ranked above a common but cosmetic one. I'll put that list together with rough effort estimates from the dev team, so the group deciding priorities can see both impact and cost. I also keep a running note of smaller issues so they don't get lost even if they're not first in line."

5. How do you document your designs so developers can build them accurately without constant back-and-forth?

Why they ask

Handoff quality directly affects build time and rework, so they're checking whether your documentation habits match how their engineering team works.

How to structure your answer

Walk through your documentation approach step by step: what you include, which tools you use, and how you handle edge cases and states that aren't obvious from the main screens.

Example answer

"I keep a design system file in Figma with named components and states, so developers aren't guessing at hover, error or empty states. For anything with logic, like conditional fields or validation rules, I write short annotations directly next to the relevant screen rather than a separate long document nobody reads. I also try to sit in on the first build review so I can catch misunderstandings early instead of after the feature's already coded."