

Software Engineer interview questions

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

What to expect

Software Engineer interviews in Australia typically combine a technical screen with behavioural and system design discussions. Expect a mix of live coding, architecture reasoning, and questions about how you work within a team, since most roles now sit inside Agile squads with product and design stakeholders.

- **Technical/coding:** Live or take-home problems testing your ability to write correct, efficient code in languages like Python, Java, TypeScript or Go.
- **System design:** Open-ended questions about designing architecture or database schemas for scalability and performance.
- **Behavioural:** Questions about past project work, collaboration with product managers, and handling disagreement or ambiguity.
- **Debugging/scenario:** Judgement-based questions about diagnosing production issues and deciding how to respond under pressure.
- **Process:** Questions about how you run testing, code review, and CI/CD practices day to day.

Most processes run two to four stages: an initial recruiter or hiring manager screen, a technical interview (coding exercise or take-home followed by a review), a system design or architecture discussion, and a final round covering culture fit and team collaboration. Larger employers may split coding and system design into separate interviews with different panel members.

1. Walk me through how you'd design a database schema for a system that needs to scale to a large number of concurrent users.

Why they ask

Tests your grasp of system architecture and database design under real performance and scalability constraints, a core part of the role.

How to structure your answer

Process walk-through: clarify requirements first, outline entities and relationships, discuss indexing and normalisation trade-offs, then address scaling strategies (sharding, caching, read replicas) and how you'd validate the design.

Example answer

"I'd start by clarifying read/write patterns and expected data volume, since that changes the design significantly. I'd sketch the core entities and relationships, decide where normalisation helps versus where denormalising for read performance makes sense, and add indexes on the fields we'll query most. For scale, I'd look at caching frequently read data, and if write volume becomes a bottleneck, consider sharding or read replicas. I'd validate the design against a couple of the heaviest expected query patterns before committing to it."

2. Tell me about a time you had to debug a production issue under time pressure.

Why they ask

Directly reflects a listed task, debugging production issues and implementing monitoring and logging, and reveals how you handle pressure and incomplete information.

How to structure your answer

STAR: describe the situation, your specific task, the actions you took to diagnose and fix it, and the result including what you changed afterwards to prevent recurrence.

Example answer

"We had an intermittent failure in a payment flow that only showed up in production, not in staging. I pulled logs and traced it to a race condition triggered under high concurrent load. I reproduced it locally by simulating concurrent requests, confirmed the root cause, and shipped a fix that added proper locking around the shared resource. Afterwards I added monitoring alerts on that endpoint so we'd catch similar issues earlier next time rather than relying on customer reports."

3. A stakeholder asks for a feature that conflicts with technical best practice or introduces significant technical debt. How do you handle it?

Why they ask

Probes how you translate requirements into technical specifications while managing trade-offs with product managers and stakeholders, a task explicitly listed for this role.

How to structure your answer

Scenario/judgement: explain how you'd assess the trade-off, what you'd communicate to the stakeholder, and how you'd reach a workable compromise.

Example answer

"I'd first make sure I understand why they want it that way, since there's often a business reason I'm not seeing. I'd then explain the technical debt or risk in plain terms, not jargon, and offer an alternative that meets their goal with less long-term cost. If they still need it shipped quickly, I'd agree to it but document the debt and get a rough timeline for addressing it later, so it doesn't get forgotten."

4. How do you approach writing automated tests for a new feature?

Why they ask

Directly tied to the listed task of developing and executing automated test suites to ensure code quality and prevent regressions.

How to structure your answer

Process walk-through: describe your testing strategy from unit through integration level, what you prioritise, and how it fits into the CI/CD pipeline.

Example answer

"I start with unit tests for the core logic, focusing on edge cases rather than just the happy path. Then I add integration tests for the parts that touch the database or external services, since that's where most real bugs show up. I make sure these run automatically in the CI pipeline on every pull request, so nothing merges without passing, and I keep an eye on flaky tests since they erode trust in the suite over time."

5. Explain a technical concept you've worked with to someone without a technical background.

Why they ask

Tests written and verbal communication skills, which matter when collaborating with product managers and non-technical stakeholders.

How to structure your answer

Direct explanation demonstration: pick a real example, simplify without dumbing down, and check the interviewer's follow-up questions for clarity.

Example answer

"I once had to explain why an API rate limit was causing slow page loads to a product manager. I compared it to a shop with a limit on how many customers can be served per minute: if too many requests come in at once, some have to wait in line. That framing helped them understand why we needed to either negotiate a higher limit or batch our requests more efficiently, without getting into HTTP status codes."

6. Why are you looking to move into this role, and what draws you to our team specifically?

Why they ask

Standard behavioural/motivation question interviewers use to gauge genuine interest and cultural fit, especially relevant given the role's strong overlap with adjacent paths like back end or full stack development.

How to structure your answer

Short narrative: connect your career direction to specific aspects of the role and team, avoiding generic answers.

Example answer

"I've enjoyed the architecture and system design side of my current work more than pure feature delivery, and I'm looking for a role with more scope to influence those decisions. From what I've read about your engineering team's approach to testing and deployment, it matches the way I like to work, with strong CI/CD practices and a real emphasis on code quality rather than just shipping fast."