

QA Tester interview questions

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

What to expect

QA Tester interviews mix technical questions about how you actually test software with behavioural questions about how you handle pressure, disagreement and ambiguous requirements. Because this role sits between developers and end users, interviewers are also checking how clearly you communicate a defect and how you make judgement calls when there isn't time to test everything.

- **Process/technical:** Walk-through questions on how you'd approach testing a specific feature or write a test case from requirements.
- **Behavioural:** Past-experience questions about finding critical bugs, handling defect backlogs, or working with developers on fixes.
- **Scenario/judgement:** Hypotheticals about prioritising limited testing time or handling disagreement with a developer over a bug report.
- **Conceptual/tooling:** Questions checking your understanding of testing concepts and familiarity with tools like Selenium, JIRA and Postman.

Most QA Tester interviews open with a few questions about your testing background and the tools and industries you've worked in, move into a scenario or two testing your judgement on prioritisation and defect handling, and often include a short practical exercise such as writing test cases for a sample feature or reviewing a bug report for clarity. They usually close with questions about how you work with developers and product owners day to day.

1. Walk me through how you'd approach testing a new login feature before release.

Why they ask

This checks whether you think systematically about coverage rather than testing randomly, and whether you consider security and edge cases without being prompted.

How to structure your answer

Answer as a step-by-step walkthrough: start with reviewing requirements, then outline test case design (happy path, edge cases, negative cases), execution, defect logging, and regression checks after fixes.

Example answer

"I'd start by reviewing the requirements and any acceptance criteria to understand expected behaviour, including password rules and lockout thresholds. I'd design test cases covering the happy path, invalid credentials, account lockout after repeated failed attempts, and session handling. I'd also check for basic security issues like whether error messages reveal too much information. After executing the cases manually first, I'd automate the stable ones in Selenium for regression, log any defects in JIRA with clear reproduction steps, and re-test once fixes were deployed to confirm nothing else broke."

2. Tell me about a time you found a critical bug close to a release deadline.

Why they ask

Interviewers want to know you can identify severity accurately and communicate risk without causing unnecessary panic or, worse, letting a serious issue slide to hit a deadline.

How to structure your answer

Use STAR: situation, task, action, result, focusing on how you assessed severity and who you escalated to.

Example answer

"During regression testing ahead of a release, I found that a discount code could be applied multiple times to the same order, which affected the checkout total. The release was scheduled for the next day. I logged the defect in JIRA as high priority with the reproduction steps and screenshots, then flagged it directly to the lead developer and product owner rather than waiting for the daily stand-up. We agreed to delay that specific feature behind a flag while the rest of the release went ahead. The fix was verified and the feature was switched on two days later with no further issues."

3. How do you prioritise what to test when you don't have time to test everything before a release?

Why they ask

This is a common real constraint in QA work, and the answer reveals whether you can make a risk-based call rather than testing everything equally or panicking under time pressure.

How to structure your answer

Answer as a judgement-under-pressure response: state the principle you use to prioritise, then give a concrete example of applying it.

Example answer

"I prioritise based on risk and usage, focusing first on core user flows like checkout or account access, then on areas that changed most recently since that's where regressions are most likely. Low-traffic or cosmetic features get tested last or covered by smoke tests only. On one release with a compressed timeline, I focused testing on the payment flow and the three most-used pages, and flagged to the team that the admin reporting section hadn't been fully regression tested so they could make an informed call on the risk."

4. What's the difference between manual and automated testing, and how do you decide which to use?

Why they ask

This checks your practical understanding of testing strategy rather than just tool familiarity, which matters given the role uses both Selenium/TestNG and manual test execution.

How to structure your answer

Answer as a direct conceptual explanation with a worked example, not a story.

Example answer

"Manual testing is better for exploratory work, usability checks, and anything that changes frequently, since automation scripts for constantly changing features become a maintenance burden. Automated testing works best for stable, repeatable checks like regression suites, where the same steps need to run every build. In practice, I test new features manually first to understand actual behaviour, then automate the ones that pass and are unlikely to change soon, using TestNG to structure the suite and run it as part of the build pipeline."

5. A developer tells you a defect you've logged isn't actually a bug. How do you handle that?

Why they ask

Tests your interpersonal judgement and whether you can hold your position on evidence without becoming defensive or backing down when you shouldn't.

How to structure your answer

Scenario/judgement structure: describe how you'd verify, communicate, and resolve the disagreement.

Example answer

"I'd first go back to the requirements or expected behaviour to check I hadn't misunderstood the intended functionality. If my reproduction steps still held up, I'd walk the developer through them directly, showing the exact conditions and result rather than just restating the ticket. Sometimes it

comes down to different assumptions about intended behaviour, in which case I'd loop in the product owner to clarify what's actually expected. I try to keep it focused on the evidence rather than who's right, since the goal is getting the product working correctly."

6. How do you write a test case from a set of business requirements?

Why they ask

This is a core daily task for the role, and interviewers want to see your method for translating requirements into concrete, testable steps.

How to structure your answer

Answer as a process walkthrough covering inputs, method, and output.

Example answer

"I start by reading through the requirements and any user stories to identify the expected behaviour and acceptance criteria. For each requirement, I write test cases covering the main path, boundary conditions, and likely negative scenarios, including preconditions, steps, and expected results. If requirements are vague or missing edge cases, I raise questions with the business analyst or product owner before finalising the test cases, since ambiguous requirements are one of the most common sources of gaps in test coverage."