

96 QA interview questions to hire top engineers

Questions

1. What is software quality assurance and why is it important?
2. Can you describe the software development lifecycle (SDLC) in simple terms?
3. What are some different testing types you know?
4. What is a test case, and what are the key components of a good test case?
5. What is the difference between verification and validation?
6. Explain the difference between a bug, defect, and error.
7. What is a test plan, and why is it necessary?
8. What is positive and negative testing?
9. Explain black box testing in layman's terms.
10. Explain white box testing in layman's terms.
11. What is regression testing, and when is it performed?
12. What is the purpose of exploratory testing?
13. What are some bug tracking tools you are familiar with?
14. How would you report a bug effectively?
15. What are some key qualities of a good QA tester?
16. If you found a bug right before a release, what would you do?
17. What does it mean for a test to 'pass' or 'fail'?
18. How do you prioritize testing tasks?
19. What is the importance of documentation in QA?
20. What is the difference between manual and automated testing? What are the pros and cons of each?
21. Explain what is meant by 'test coverage'.
22. Have you ever worked in an Agile environment? If so, how does QA fit into the Agile process?
23. Can you describe a situation where you had to advocate for quality, even when it was unpopular?
24. How do you stay up-to-date with the latest trends and technologies in QA?
25. Imagine you are testing an e-commerce website. What are some of the first things you would test?
26. What are some different levels of software testing?
27. How do you approach testing a feature that has incomplete or changing requirements?
28. Describe a time you had to advocate for quality when others didn't see the need. What was the outcome?
29. Explain the difference between black box, white box, and gray box testing. Provide examples of when you'd use each.
30. How do you prioritize testing efforts when you have limited time and resources?
31. What are some common challenges you've faced in test automation, and how did you overcome them?
32. Explain the concept of test data management. Why is it important, and how do you ensure its effectiveness?
33. Describe your experience with different testing methodologies (e.g., Agile, Waterfall). Which do you prefer and why?
34. How do you handle conflicting feedback from different stakeholders during the testing process?
35. Explain the importance of root cause analysis in software testing. Walk me through your process of identifying the root cause of a defect.
36. What are some strategies for testing APIs? What tools do you use?
37. How do you ensure the test environment accurately reflects the production environment?
38. Describe your experience with performance testing. What metrics do you focus on, and what tools have you used?
39. How do you approach testing for security vulnerabilities? What are some common security testing techniques?
40. Explain the difference between verification and validation. Provide examples of each in the context of software testing.
41. How do you measure the effectiveness of your testing efforts? What metrics do you use to track progress and identify areas for improvement?
42. Describe a time when you found a critical bug late in the development cycle. How did you handle the situation?
43. How do you stay up-to-date with the latest trends and technologies in software testing?
44. Explain the concept of continuous integration and continuous delivery (CI/CD) and how testing fits into that process.
45. How do you determine the appropriate level of test coverage for a given project?
46. Discuss your experience with mobile testing. What are some specific challenges and considerations for testing mobile applications?
47. How would you approach testing a feature that has no formal specifications?
48. Describe a time you had to advocate for quality when it wasn't a priority. What did you do?
49. Explain how you would design a test automation framework from scratch. What are the key components?
50. Let's say a critical bug slips into production. What steps do you take to analyze the root cause and prevent it in the future?
51. How do you handle conflicting priorities between testing speed and test coverage?
52. How would you test a complex algorithm without a clear understanding of the expected output?
53. Describe your experience with performance testing and identifying bottlenecks in a system.
54. How would you approach testing a machine learning model for bias and fairness?
55. Explain how you stay up-to-date with the latest trends and technologies in software testing.
56. Describe a situation where you had to challenge a developer's decision regarding code quality or functionality. What was the outcome?
57. How do you measure the effectiveness of your testing efforts?
58. Explain how you would test a highly scalable and distributed system.
59. What strategies do you use for testing APIs, and what tools do you find helpful?
60. How would you go about testing a mobile application under various network conditions?
61. Describe your experience with security testing and identifying vulnerabilities in software.
62. How do you integrate testing into a CI/CD pipeline to ensure continuous quality?
63. Explain how you would test a system that interacts with multiple third-party services.
64. How would you handle a situation where you are constantly finding bugs, but they are never being fixed?
65. Describe a testing project you are particularly proud of, and explain why.
66. How do you approach testing a system for accessibility, ensuring it meets the needs of users with disabilities?
67. Explain how you would test a system that needs to handle a large volume of data.
68. How do you prioritize test cases when you have limited time and resources?
69. Describe your experience with different testing methodologies, such as Agile, Waterfall, and DevOps.
70. How would you test a feature that is constantly changing based on user feedback?
71. Explain how you would test a system for internationalization and localization.
72. Describe a time when you had to advocate for quality against pressure to release quickly. What was the outcome?
73. How would you design a test automation framework from scratch for a complex e-commerce application?
74. Explain the difference between various types of performance testing (load, stress, endurance) and when you would use each.
75. Describe a situation where you identified a critical bug in production. What steps did you take to mitigate the impact?
76. How do you stay up-to-date with the latest trends and technologies in software quality assurance?
77. Explain the importance of non-functional testing and give examples of different types of non-functional tests.
78. How do you approach testing a microservices architecture?
79. Describe your experience with different bug tracking tools and how you use them effectively.
80. How do you measure the success of a QA team?
81. What is your approach to testing AI/ML-powered applications?
82. Explain the concept of test-driven development (TDD) and its benefits.
83. Describe a challenging testing project you worked on and the lessons you learned.
84. How do you prioritize testing efforts when resources are limited?
85. Explain the different levels of testing (unit, integration, system, acceptance) and their purpose.
86. How do you handle conflicting priorities between testing and development teams?
87. What are the key considerations for testing mobile applications on different devices and platforms?
88. Describe your experience with security testing and common security vulnerabilities.
89. How do you ensure that accessibility testing is incorporated into the software development lifecycle?
90. Explain the role of data in testing and how you use data to create effective test cases.
91. How do you collaborate with developers to improve code quality?
92. What are your thoughts on the future of software testing and the impact of emerging technologies?
93. Explain the concept of continuous integration and continuous delivery (CI/CD) and how it affects testing.
94. How do you approach testing a system that integrates with third-party APIs?