



Technical Interviews & Assessments for QA/Testers

We know that Quality Assurance and Software Testing are not the same. Both Quality Assurance and Quality Control as demonstrated by Software Testers are essential to deliver high-quality products to users. Often companies require one to possess both QA and Software Testing skill sets.

When demonstrating QA capabilities in an interview, it is important to show that you are process oriented. It is all about preventing defects by ensuring the processes used to manage and create deliverables works. Not only does the process work, but is consistently followed by the team. You want to demonstrate that you design and build engineering processes that assure quality is achieved in an effective and efficient way.

When demonstrating your capabilities as a Software Test Engineer, you need to show that you are product oriented. You want to demonstrate your ability to detect bugs and inconsistencies etc. by inspecting and testing the product. This involves checking the product against a predetermined set of requirements and validating that the product meets those requirements.

Ultimately you want to demonstrate your ability to be proactive (QA) to ensure the assembly line actually works and reactive (Testing) to ensure the products coming off the assembly line are checked to verify they meet the required specifications.

Technical Assessments

During the technical parts of the interview process, hiring managers are trying to evaluate the following:

- Knowledge of Testing Fundamentals
- Test Design Techniques
- Knowledge and understanding of the Software Development Lifecycle
- Technical ability

The most common form of assessing for these skills are through:

- **Take-Home or Online Technical Assessments**

The most important thing to understand when doing a take-home assessment is to know the application of the test before starting to test it. Ideally, you do not only want to look at the requirement document. If possible, ask for and review the architecture doc, design doc, sequence diagram, and activity flow diagram. These will give you extra context that may lead to choosing a different and better technique or concept of testing.

While the content of a QA/Software Testing technical assessment is vastly different, the challenges you encounter will be similar to those of a Software Engineer technical assessment. Hiring managers could use platforms such as Hackerank, Codility, TestDome or InterviewMocha. The [Technical Interview Prep PDF](#) will be a useful resource for online assessments.

- **Technical Interviews**

The key to acing a technical testing interview is to demonstrate knowledge of Testing Concepts. One needs to be very good at this, especially the Manual Testing methodologies. But only knowing different testing concepts is half of the work.

The next most important thing is to know which type/technique/concept of testing can be applied at what stage of the SDLC. Being able to answer “What should I test and when?” is very important. There might be some concepts that you use in your current role or that you are very familiar with, which do not apply to a particular context of the company you are interviewing with. That’s why it’s always better to have an idea of all testing practices e.g. Localization Testing, Time Zone Testing, etc. Knowing more than what you have worked on will help you better answer the different questions from the interviewer.

Aside from “curiosity to learn” you can upgrade your skill in the following areas:

- Some hands-on basic Database/SQL queries and concepts.
- Any basic scripting language (for Automation Testing).
- Networking and system administration concepts will help you in system domain projects.

There are five themes of questions you will likely be asked:

1. *Questions about previous projects*

Draw on your testing experiences and knowledge to demonstrate your ability. Remember that if you've mentioned something on your CV, interviewers will expect you to be able to demonstrate expertise in an interview.

2. *Questions about tools*

Demonstrate your competence and ability by using examples from your experience. For example: your experience working with Selenium on project XYZ.

3. *Questions about your behaviour and how you work with others*

Testing/QA requires collaboration and good communication skills. Since, at times, you need to deliver constructive criticism to fix or enhance the product, it's important to know how you go about delivering not so great news to colleagues and counterparts.

4. *Questions about your testing approach*

Don't feel you need to follow the traditional methods. Demonstrate implementing new things in testing. Think from a user perspective. Think about how the user will use your application.

5. *Questions about your testing terminology knowledge*

Make sure you review testing fundamentals and computer science basics. Review core concepts and theories that are essential to good practice.

Some important things to know:

- The Seven Principles of Software Testing
 1. Testing shows presence of defects
 2. Exhaustive testing is not possible
 3. Early testing
 4. Defect clustering
 5. Pesticide paradox
 6. Testing is context dependent
 7. Absence of errors fallacy
- The approach to QA/Testing into core Development Methodologies like Waterfall Model, Iterative Development, Agile Methodology and Extreme Programming
- The phases of the Software Testing Life Cycle
- Manual Testing vs Automation Testing
- The objectives, characteristics, parameters and types of Functional, Non-Functional and Maintenance Testing to inform in which context you would implement different types of testing.
- Writing TestCases
- Test Documentation Best Practices

Commonly Asked Questions

Quality Assurance:

1. Compare a test plan to a test case. What does each include?
2. What is a software development life cycle?
3. What is positive testing? Negative testing? Compare the two.
4. What is a primary key? What is a unique key? How are they different?
5. What is the difference between requirements and specifications?
6. When do we perform functional testing?
7. What is the sequence to write a test case?
8. What is an application programming interface?
9. What if there isn't enough time for thorough testing? What would you do?
10. What is the difference between performance testing and load testing?

Test Engineering:

1. How do you distinguish a symptom from a cause in testing?
2. Describe a test strategy which you have found to be most effective.
3. What test cases would you write for a remote desktop feature?
4. Do you prefer a greater or lesser degree of documentation during testing? Please explain your reasoning.
5. Please describe a time when you successfully collaborated with developers to deploy an application or product.
6. What are the three different methods of testing?
7. Explain the bug life cycle or defect life cycle.
8. What is verification and validation in software testing?
9. Name the basic components of a defect report format.
10. What is meant by defect cascading?

Blog Posts

Check out these articles from our blog – they may be useful for you in preparing for your interview processes and beyond 🎉

- [The Changing Face of Software Testing: Where Do We Go From Here?](#)
- [Simulating The Right Thing: Property-Based Testing](#)
- [A Tester's Guide to Unit Testing](#)
- [Sedan to Supercar - Test Always!](#)

Resources:

- <https://www.softwaretestinghelp.com/how-to-prepare-for-software-testing-interview/>
- <https://www.stickyminds.com/article/how-ace-your-next-qa-job-interview>
- <https://mashable.com/2011/11/06/how-to-ace-technical-interview/>
- <http://www.base36.com/2013/06/qa-interview-questions/>
- <https://www.functionize.com/blog/quality-assurance-vs-quality-control-vs-testing-whats-the-difference/>
- <https://www.lever.co/blog/test-engineer-interview-questions/>
- <https://assurity.nz/archives/how-to-prepare-for-a-software-testing-interview/>
- <https://www.guru99.com/testing-methodology.html>
- <https://www.guru99.com/non-functional-testing.html>