50+ QA Interview Questions And Answers 2025

Prepare for your next interview with this list of QA interview questions and answers. The list includes interview questions for both QA testers and managers.

Download Studio View demo

50+ QA Interview Questions And Answers 2025

Katalon Team

Contributors 16 min read Updated on Nov 30, 2025

Interviews are anxiety-inducing, and nothing can better boost your confidence than solid preparation. Here are 50+ QA interview questions and answers to help you review key concepts in Quality Assurance and testing.

We've categorized these questions by difficulty, type, and job role (QA manager, QA lead, QA tester). You can read through all questions or jump to the most relevant sections.

Smart Summary

Prepare confidently for QA interviews by mastering over 50 essential questions and answers. Our resource categorizes insights by difficulty, type, and role, providing the foundational knowledge necessary to demonstrate expertise in quality assurance and testing processes.

- **Strategic Test Planning Models:** Utilize risk-based, model-based, or hybrid approaches to align testing efforts with project needs, prioritizing high-impact areas or systematically designing tests based on system visualizations.
- **Prioritize Test Case Execution Effectively:** Apply nine key criteria, including business impact, risk, frequency of use,

Read more ⌄

Common QA Interview Questions and Answers

1. What is Quality Assurance? Give a real-life example in software development.

QA in software development ensures software meets quality standards by testing functionality, performance, usability, and security.

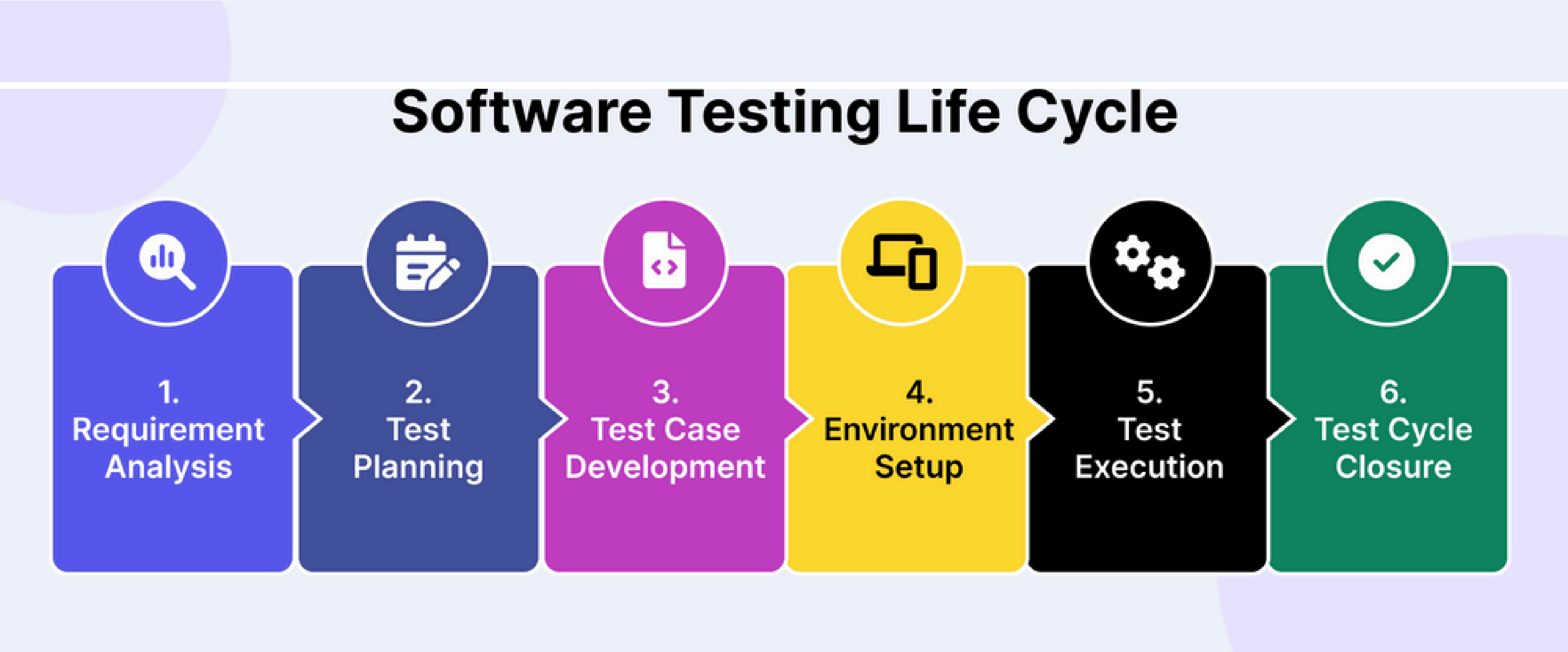
For example, before launching a mobile banking app, the QA team checks whether users can log in, view balances, transfer funds, and make payments. They also test backend module communication.

If bugs are found, testers report them for fixing. Once resolved, QA retests to confirm fixes and ensure no regressions.

> 📚 **Read More:** Software Quality Management Best Practices | 5 Do's & Don'ts

2. What is the Software Testing Life Cycle? Explain each step.

The software testing life cycle is a standard process teams follow to ensure software is fully tested and meets quality standards.



1. Requirements Analysis

- QA analyzes requirements to create a test plan.
- Developers translate requirements into code.
- Testers break requirements into test cases.
- Teams collaborate to assess feasibility.

2. Test Planning

- Align on testing strategy.
- Define test objectives (functionality, usability, security, etc.).
- Outline test scope (what will/will not be tested).

- Allocate resources and environments.
- Set timelines and milestones.
- Choose test approaches (manual, automation).

3. Test Case Development

- Write test cases based on scenarios.
- Use tools like Xray for manual testing.
- Automate tests with Katalon, Ranorex, or Selenium.

4. Test Execution

- Run tests manually or via automation.
- Set up testing environments (cloud or physical devices).
- Identify, track, and report defects.

5. Test Cycle Closure

- Review test reports, unresolved defects, and release readiness.

3. What is your experience with automation testing tools?

An automation testing tool helps teams write, run, debug, and report test scripts efficiently. It should also integrate with DevOps tools such as test management, defect tracking, CI/CD systems, and container platforms.

Test automation tools are categorized based on:

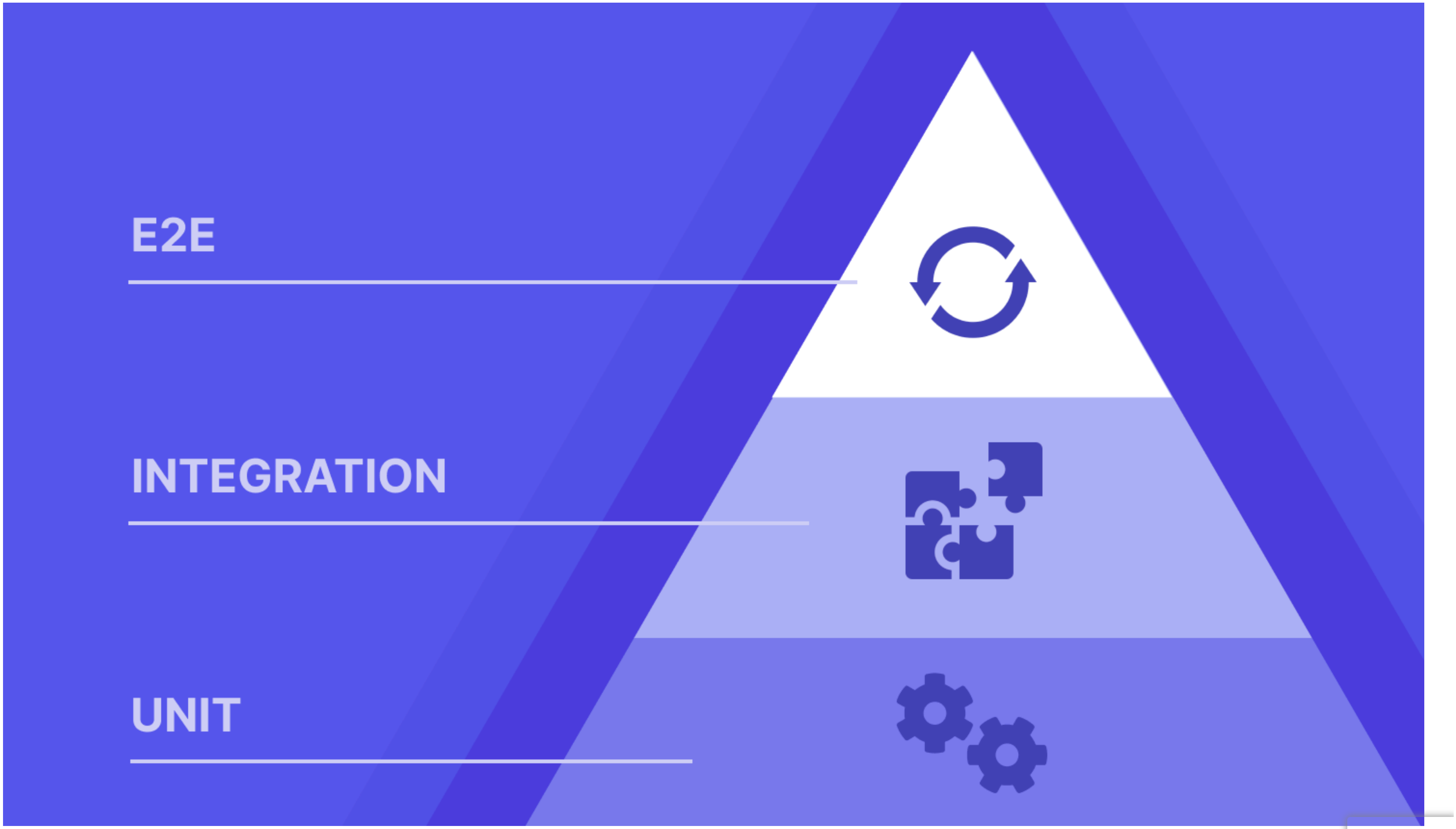
- **Application Type**: Web, mobile, API, or desktop applications.
- **Application Layer**: UI testing vs. API testing.

Some of the common names by application types are:

- API and web services testing: Postman, Katalon, SoapUI

- Web UI testing: Katalon, Ranorex, TestComplete, Cypress
- Mobile testing: Appium, XCUITest, Katalon, TestComplete

4. Explain the different test levels and give examples



Testing levels refer to the different stages and granularity of testing.

Let’s use an ecommerce website to give sample scenarios for **unit testing, integration testing, and end-to-end testing.**

- Unit testing: tests the login page to handle valid/invalid credentials.
- Integration testing : tests the payment processing flow where PayPal verifies card details.
- End-to-end testing: tests the full order-placement workflow (inventory update → confirmation email → payment processing → order status update).

5. What is your approach to test planning?

We can approach test planning through three major methods: **risk-based, model-based, or hybrid.**

Risk-Based Test Planning

- Testing is prioritized based on potential impact and likelihood of defects.
- High-risk areas receive deeper testing; low-risk areas receive lighter coverage.
- *Example:* A banking app prioritizes security and transaction features over UI cosmetics.

> 📚 Read More: Risk-based approach to Regression Testing: A simple guide

Model-Based Test Planning

- Uses models (diagrams, workflows, state transitions) to design and automate tests.
- Helps visualize complex systems and generate cases systematically.
- *Example:* A chatbot is tested using a state-transition model to cover conversation paths.

Hybrid Test Planning

- Combines multiple approaches (e.g., risk-based + model-based) to optimize coverage.
- Provides flexibility for diverse project needs.
- *Example:* A healthcare app uses risk-based testing for security and model-based testing for patient workflows.

> 📚 Read More: Hybrid testing: How to fuse manual with automation testing?

6. What is exploratory testing?

Exploratory testing is a testing approach that involves simultaneous learning, test design, and execution. It is used when there is no formal test plan or script, and when teams need to discover issues not yet covered by existing test cases.

It is typically performed by experienced testers who rely on domain knowledge, intuition, and creativity to uncover defects.

If you're interested in exploratory testing, Callum Akehurst has written a great blog on how exploratory testing drives good testing.

7. Explain stress testing, load testing, and volume testing

Stress Testing – Pushes the application beyond its limits to see how it breaks. This helps developers prepare for failures and improve system resilience.

Load Testing – Tests the system under expected user traffic to identify performance issues such as slow response times or high CPU usage.

Volume Testing – Evaluates how well the system processes large amounts of data to ensure no data loss or corruption.

> 📚 Read More: Performance Testing vs Load Testing: A Complete Guide

8. What is Agile testing and why is it important?

Agile testing is a testing approach aligned with the Agile methodology, which emphasizes collaboration, continuous feedback, and rapid iteration.

In Agile testing, testing is integrated into development and performed iteratively throughout the lifecycle. Developers, testers, and stakeholders work together to ensure the product meets customer requirements and maintains high quality.

The importance of Agile testing lies in its ability to catch defects early, provide continuous validation, and enable quick adaptation to changing requirements and feedback.

> 📚 Read More: How to do regression testing in Agile teams?

9. What is the difference between TDD vs BDD?

TDD (Test-Driven Development) is a coding method where developers write tests before writing the actual code, resulting in cleaner and more maintainable software.

BDD (Behavior-Driven Development) focuses on defining software behavior from the end-user perspective using plain language that both technical and non-technical stakeholders understand.

Below is a table for quick comparison:

	TDD	BDD
Definition	Start software development by writing test cases.	Use given-when-then syntax to: • Define software features and functionalities • Write scenarios, step definitions, and automated tests • Write BDD automated tests
Goal	Test coverage and code testability	Alignment between technical and business stakeholders
Test writing	Developers	Varies by team. Ideally: • Testers and business analysts write BDD and acceptance tests • Developers and automation engineers implement the tests
Tools	Test libraries: JUnit, NUnit, TestNG, Selenium Testing tools: Katalon, TestComplete	Available frameworks: Cucumber, SpecFlow, Behave

> 📚 **Read More:** TDD vs BDD: A Comparison

10. What is Data-driven Testing?

Data-driven testing is a design pattern that reuses the same test flow across multiple sets of input data.

Scenario	Test Case	Data Input
Login Scenario 1. Enter username 2. Enter password 3. Click *Login*	Test Case 1	Valid username and password combinations
	Test Case 2	Invalid username and password combinations

The purpose of data-driven testing is to avoid hard-coding single input values. Instead, tests are parameterized and read data dynamically from sources such as databases, spreadsheets, or XML files.

Data-driven testing is particularly useful for:

- Input validation
- Boundary testing
- Error handling and exception testing
- Compatibility testing across different browsers, devices, and OS configurations
- Performance testing with varying data loads

> 📚 **Read More:** A Guide to Data-driven Testing

11. What is performance testing?

Performance testing evaluates a system's responsiveness, scalability, stability, and speed under different workload conditions. It determines how the application behaves under normal and peak load, such as high traffic, large data volumes, or concurrent user interactions.

The results help identify bottlenecks, optimize performance, and improve overall user experience.

12. What is accessibility testing?

Accessibility testing evaluates whether a software application or website is usable by all users, including people with visual, auditory, motor, or cognitive disabilities. It checks compatibility with assistive technologies such as screen readers, magnifiers, and voice recognition tools.

13. Compare manual testing vs automated testing — Should teams move to automation?

Aspect	Manual Testing	Automated Testing
Definition	Testers manually perform the actions in the application. Tests are typically written in text editors, Xray, test management tools, or spreadsheets.	Testers define the interaction steps, then write automation scripts to execute them. Scripts can run on-demand or on schedule.
Cost	Lower upfront cost; depends on human testers. Difficult to scale.	Investment in developers/automation engineers and tools (automation frameworks, CI, test management, defect tracking).
Test Coverage	Low	High
Reusability	Test content cannot be reused easily.	Test content is highly reusable, including: • Test cases • Variables • Action keywords • Custom keywords
Types of Testing	Exploratory testing Usability testing Ad hoc testing	Regression testing Integration testing Data-driven testing Performance testing

Tester	Business stakeholders Manual test engineers	Developers or automation engineers

If testing is repetitive and requires frequent regression cycles, teams should consider automation. Manual testing still adds value for exploratory or ad-hoc scenarios. The decision depends on project type, goals, and complexity. Here is a guide to move from manual testing to automation.

> 📚 **Read More:** 15 Different Types of QA Testing

14. Compare black-box testing vs white-box testing

	Black-box Testing	White-box Testing
Definition	Write tests **without visibility** into internal code or structure.	Write tests **with full visibility** into internal code and structure.
Goal	User experience, security, compatibility.	Code quality, logic correctness, optimization.
Testing levels	UI end-to-end testing Compatibility testing	Unit testing Integration testing Static code analysis
Tester	Business stakeholders Manual test engineers	Developers

15. Explain end to end testing in your own words. Compare End to End Testing vs Integration Testing

End-to-End testing checks the entire application to ensure all parts work together as expected, just like a real user would experience. It tests everything from the front end to the back end, including databases, APIs, and third-party services.

Aspect	End-to-End Testing	Integration Testing
Scope	Tests the entire system from start to finish.	Checks how different modules work together.
Purpose	Ensures the full application functions correctly.	Verifies data flow between connected components.
Example	Testing a complete online shopping process.	Checking if the payment gateway communicates with the checkout page.

Top QA Tester Interview Questions And Answers

The list above includes common QA interview questions that anyone in the industry may face. This section provides QA interview questions specifically tailored for QA testers.

QA testers are responsible for executing test cases, identifying and documenting defects, and providing feedback to developers. They are often asked technical questions to assess their understanding of testing processes and automation best practices.

16. How do you perform visual testing?

Visual testing can be performed manually, where testers visually inspect the application for UI inconsistencies. However, this method can be time-consuming and prone to human error.

Many testers use Image Comparison techniques: they capture baseline screenshots of UI elements and compare them with new screenshots to detect unexpected changes.

Even so, this approach may generate false positives. Using visual regression testing tools helps reduce false positives and improves efficiency.

> **Read More:** How Automated Visual Testing Will Redefine the Testing Pyramid?

17. How do you prioritize test cases for execution?

There are many factors to consider when prioritizing test cases. Below are 9 of the most common criteria used by QA professionals:

- Business impact — test cases covering critical flows (e.g., Login, Checkout)
- Risk — areas with high potential for failure
- Frequency of use — highly used features
- Dependencies
- Complexity
- Customer or user feedback — addressing reported pain points
- Compliance requirements — tests required by industry regulations
- Historical data — tests that frequently reveal defects
- Test case age — older tests may need maintenance

> 📚 **Read More: How to select test cases for automation?**

18. What are the key components of a good test case?

- Know when manual or automated testing is most appropriate
- Use suitable design methods (e.g., data-driven testing) based on requirements
- Ensure test cases are scalable and reusable across cycles or projects
- Assign each test case a unique name or identifier for easier filtering and regression runs
- Include both positive and negative test data to uncover edge cases
- Run tests in environments (browser/device/OS) most used by end-users
- Store test cases in a centralized repository or test management tool for easy access and maintenance
- Design independent test cases so execution order does not impact results

19. What are defect triage meetings?

Defect triage meetings are used to prioritize and assign defects to the right team members. During these meetings, QA testers present identified defects, including severity, priority, and potential impact, to help determine next actions.

20. Can you provide an example of a particularly challenging defect you identified and resolved?

There is no single correct answer. It depends on your personal experience. You can use the following framework to structure your response:

- **Step 1:** Describe the defect in detail and explain how it was discovered (e.g., testing, user reports).
- **Step 2:** Explain why the defect was especially challenging.
- **Step 3:** Outline the actions you took to resolve it, including collaboration with developers or others.
- **Step 4:** Discuss obstacles or roadblocks faced during resolution.
- **Step 5:** Explain how you verified the fix and the impact of resolving the issue.
- **Step 6:** Reflect on what you learned from the experience.

21. Explain API Testing and show your approach to API Testing

API testing is the process of verifying that an API behaves as expected. It checks functionality, performance, security, and how the API handles various inputs and edge cases.

Key considerations when designing API tests:

- Read the API documentation to understand functionality and technical requirements
- Consider the architectural style — REST, GraphQL, and SOAP require different testing approaches
- Automate data-driven tests to validate flows against varied data types, formats, and scenarios
- Manage endpoints by grouping them to avoid duplicates and ensure complete scenario coverage

22. How do you ensure that test cases are comprehensive and cover all possible scenarios?

It’s impossible to cover every scenario, but testers should aim to expand beyond the happy path, which tests the system under ideal conditions.

In addition to standard cases, testers should include edge cases and negative scenarios — unusual inputs, unexpected user behavior, and invalid data. These areas are more likely to expose vulnerabilities or defects, improving overall test coverage.

23. What is your approach to identifying and reporting defects?

Many QA testers follow this workflow when identifying and reporting defects:

1. Replicate the defect and collect details such as steps to reproduce, screenshots, logs, and environment info
2. Assign a severity level based on the issue’s impact
3. Log the defect with clear descriptions, expected vs. actual results, and reproduction steps
4. Communicate and collaborate with developers to determine root cause and solutions
5. Follow up until the defect is fixed and verified

24. How do you measure the effectiveness of your testing efforts?

Common testing metrics include:

1. Test case execution rate
2. Test coverage
3. Defect density
4. Defect rejection rate
5. Mean time to failure (MTTF)

25. What are test management tools?

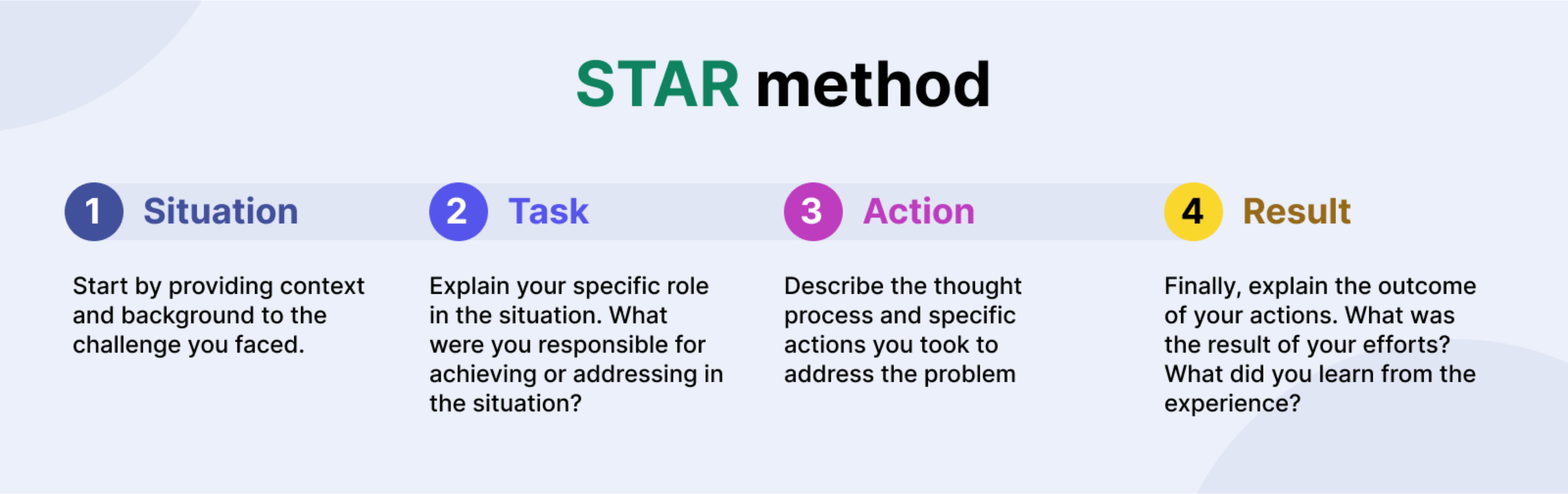
Test management tools help QA teams organize and manage their testing efforts. They support test case creation, execution tracking, test planning, reporting, and overall test lifecycle management.

Top QA Manager Interview Questions

QA interview questions for managers focus on leadership, strategy, and team management. In regulated industries like BFSI or Healthcare, compliance knowledge is also crucial. For technical QA questions, refer to the previous section. This section highlights management-focused questions.

26. Describe a situation where you had to make a difficult decision in managing a testing team, and how you handled it.

This is a situational question with no single correct answer. Use the STAR method to structure your response:



STAR method

1 Situation

Start by providing context and background to the challenge you faced.

2 Task

Explain your specific role in the situation. What were you responsible for achieving or addressing in the situation?

3 Action

Describe the thought process and specific actions you took to address the problem

4 Result

Finally, explain the outcome of your actions. What was the result of your efforts? What did you learn from the experience?

27. How do you ensure that the testing team is aligned with the development team and the product roadmap?

- Regular communication between testing and development teams
- Collaborative refinement of user stories
- Alignment on standardized testing methodologies
- Integrating testing activities into the development workflow
- Adjusting testing priorities based on roadmap or development changes

28. What is your experience with implementing an automation testing tool?

1. Identify automation opportunities and prioritize based on impact
2. Evaluate and select automation tools based on technical needs
3. Define KPIs and success criteria, and establish monitoring/reporting
4. Train the team and set up governance, policies, and procedures
5. Track effectiveness and refine the strategy as needed

> 📚 **Read More:** How To Choose The Right Automation Testing Tool?

29. How do you leverage your technical knowledge and experience to guide your team in identifying and resolving complex testing issues and challenges?

QA Managers are not just managers — they were testers first. Their technical expertise helps them guide the team through roadblocks and collaborate with development and product teams. They also perform high-level analysis and make data-driven decisions to improve testing efficiency and quality.

30. How do you manage your QA team?

This question explores your management style. Effective QA managers communicate well, show empathy, lead diverse groups, and take accountability for overall team performance.

General QA Interview Questions

Interviewers often ask general questions to understand your personality, motivations, and knowledge of the company. These questions are straightforward and can usually be answered on the spot. Review the common examples and prepare accordingly.

1. What are your key strengths? Also, share a weakness and explain how you plan to address it.
2. How did you learn about this job opportunity?
3. Why are you interested in this position?

4. What is your ideal job?
5. Describe yourself using three adjectives.
6. What do you enjoy doing outside of work?
7. Why should we hire you as a QA tester/QA analyst/QA manager?
8. What type of work environment do you prefer?
9. Who has influenced your career the most?
10. What are your career goals for the next five years?

QA Interview Questions On Background And Work Experience

These questions help interviewers understand your professional background in depth. They usually come after general questions. Provide detailed, thoughtful answers that demonstrate your expertise and show who you are in a work environment.

1. Can you tell us about your background and experience in QA?
2. What brought you to a career in QA?
3. What do you think were your biggest achievements in your QA career?
4. Have you worked on any challenging QA projects? Can you describe them in detail?
5. Can you show us your thought process when you resolve this specific testing problem?
6. How do you handle and prioritize multiple testing projects at once?
7. Can you give an example of a bug you found that required extensive communication with the dev team?
8. Are you familiar with any automation testing tools? How do you use them in your daily work?
9. What are some recent advancements or updates in QA technology that you know?
10. Can you describe a situation where you collaborated with developers or other teams to resolve a testing issue?

Tricky QA Interview Questions

These QA interview questions go beyond textbook knowledge—they focus on real-world problem-solving. Interviewers, especially experienced QA testers, can easily tell whether a candidate has encountered these situations firsthand.

How to Answer Effectively:

- Use the provided answers as **references**, but incorporate your **own experiences** for authenticity.
- Apply the **STAR method** (Situation, Task, Action, Result) to structure your responses.
- Be **specific** and **demonstrate your thought process**—how you approach problems matters more than just having the correct answer.

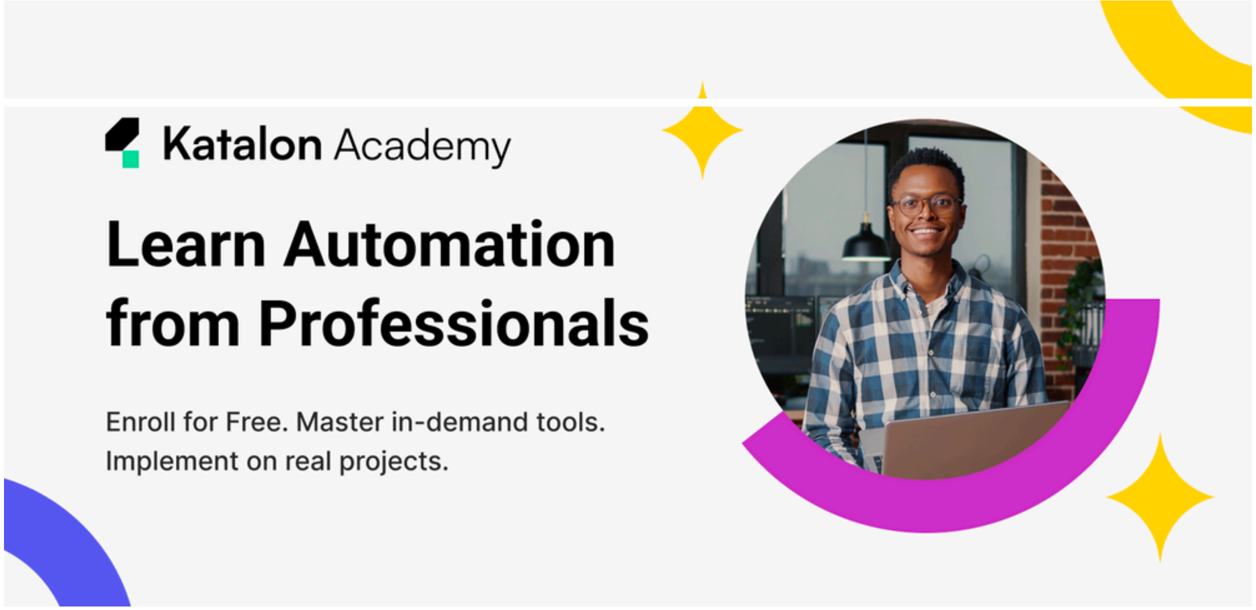
Here are some of those questions:

- What is your approach when you encounter a scenario where the requirements are incomplete or missing?
- How do you handle testing in situations where there is a tight deadline?
- Can you explain how you would test a complex software system with limited documentation?
- What are some of the most common technical problems in software testing?
- How do you ensure compatibility of a web application with multiple browsers and devices?

Recommended Readings For Your QA Interview

The list above mostly covers foundational QA theory. In many companies, you may also be asked to complete an interview project that requires demonstrating real testing skills. Explore the Katalon Blog for current insights on the testing industry, especially automation testing, which is invaluable for interviews.

As a leading automation testing platform, Katalon offers free software testing courses through Katalon Academy, a comprehensive knowledge hub full of practical learning resources.


Katalon Academy
Learn Automation from Professionals
Enroll for Free. Master in-demand tools.
Implement on real projects.

Katalon Academy provides short-form beginner courses, advanced platform guides, and specialized training for API, mobile, desktop, and web testing. The platform is updated frequently to reflect current industry practices, making it helpful even for experienced testers looking to refresh their skills.

To further prepare for interviews, explore these topic-specific question lists:

- Web API Testing Interview Questions
- Mobile Testing Interview Questions
- Software Testing Interview Questions
- DevOps Interview Questions
- Manual to Automation Testing Interview Questions
- 15 Questions to Evaluate Your QA Team's Software Testing Process
- Strategies to Deal A Software QA Engineer's Salary

50+ QA Interview Questions FAQs

What is this article about? +

Who published it and when? +

Who is this list for? +

How is the content organized and what does it cover? +

How should I use it most effectively? +

Katalon Team

Contributors

The Katalon Team is composed of a diverse group of dedicated professionals, including subject matter experts with deep domain knowledge, experienced technical writers skilled, and QA specialists who bring a practical, real-world perspective. Together, they contribute to the Katalon Blog, delivering high-quality, insightful articles that empower users to make the most of Katalon's tools and stay updated on the latest trends in test automation and software quality.

Katalon

Why Katalon

Test Creation

Test Execution

User Journey Testing

Enterprise-Grade Testing

Success Stories

Platform

Studio

Runtime Engine

TestCloud

TestOps

TrueTest

Pricing

Solutions

Visual Testing

Web Testing

API Testing

Mobile Testing

Desktop Testing

Integrations

Resources

Blog

Write for us

Community

Support

Documentation

Sample Projects on GitHub

Technical Support

Company

About Katalon

Careers

Events

Software Testing Glossary
Training and Certification
News
Webinars
Community Forum
Partners
Reports

1720 Peachtree Street NW, Suite 870, Atlanta, GA 30309

General inquiries: business@katalon.com

Legal inquiries: legal@katalon.com

Partnership inquiries: partner@katalon.com

Legal Privacy Security