Deloitte Testing Interview Guide

Comprehensive Interview Questions & Answers

Deloitte interview questions for 3-6 years of experience (2025 Edition)

SECTION 1: MANUAL TESTING — Detailed Answers

1. Explain SDLC & STLC.
SDLC defines how software is planned, developed & maintained.
Phases include: Requirement → Design → Development → Testing → Deployment → Maintenance.
STLC defines steps in testing: Requirement Analysis → Test Planning → Test Design → Environment Setup → Test Execution → Defect Reporting → Test Closure.
Key point:
"SDLC focuses on building the product, STLC focuses on validating it."

2. What is Risk-Based Testing?
Risk-Based Testing prioritizes test scenarios based on:
✔ Business impact
✔ Probability of failure
✔ Complexity
✔ Dependency on other systems
Example:
In a payment module, I prioritize card payment verification over UI alignment because financial loss risk is higher.

3. Explain Defect Life Cycle.
New → Open → Assigned → In Progress → Fixed → Retest → Reopen/Closed → Deferred/Rejected

4. Severity vs Priority — real scenario.
High Severity, Low Priority: Broken privacy policy link- The link to the privacy policy is broken and not working, which is a serious functional issue (high severity). However, since very few users click on this link, it can be addressed later (low priority).
Low Severity, High Priority: Client logo incorrect on homepage.

5. What is RTM and why is it needed?
RTM ensures all requirements → mapped to → test cases → defects.
It guarantees:
✔ 100% coverage
✔ No missed requirement
✔ Traceability during audits (important in Deloitte projects)

6. Boundary Value Analysis Example
Input field accepts 1–100

Test values: 1, 100 (valid) and 0, 101 (invalid).

7. Entry & Exit Criteria
Entry: Requirements approved, test data prepared, environment ready.
Exit: No critical/severe defects open, all planned tests executed, traceability complete.

8. Root Cause Analysis Example
Issue: Incorrect total amount on invoice.
RCA: Tax percentage was taken from outdated config table.

9.Flaky Test Handling
Flaky tests = tests that fail randomly even when nothing is wrong.
Common Causes
Timing issues / wait problems
Dependency on environment
Network delays
Unstable locators (XPath/CSS)
Deloitte Expects You to Mention
Using Explicit/Fluent waits
Improving element locators
Running tests in stable environments
Retry mechanisms (TestNG retry analyzer)
Logging and screenshots

10. KYC Edge Case Testing
Deloitte's BFSI clients heavily depend on KYC flows.
They expect strong testing of:

KYC Edge Cases
Invalid PAN / Aadhaar formats
Document blurred / rotated / cropped
Mismatch in name between documents
Expired ID proofs
Duplicate KYC attempts
Rate limits for API calls (e.g., too many OTP attempts)
Validating API error messages
Testing OCR accuracy for documents
Negative test cases for identity mismatch

SECTION 2: AUTOMATION TESTING (SELENIUM + JAVA) — Detailed Answers

1. Explain your automation framework.
A typical Deloitte answer:
Hybrid framework with:

✔ Page Object Model
✔ TestNG for execution
✔ Maven for dependency
✔ JSON/Excel for test data
✔ Log4j2 for logging
✔ Extent Reports for reporting
✔ Jenkins/Azure DevOps for CI/CD

2. Selenium 4 vs Selenium 5
Better Chrome DevTools Protocol integration
Native relative locators
Improved window & tab handling
Enhanced grid performance
Selenium 5 is more cloud-native

3. Handling dynamic elements
I use:
✔ Explicit wait (WebDriverWait)
✔ XPath with contains()
✔ JavaScript executor
✔ Stable locators like accessibility id/ARIA

4. Parallel execution in TestNG
Using:
parallel="methods" or parallel="tests" in XML.

5. Logging & Reporting
Log4j2 → captures detailed logs
Extent Reports → screenshots, step logs, environment details

SECTION 3: API TESTING — Detailed Answers

1. PUT vs PATCH vs POST
POST → create a new resource

PUT → replace the entire resource

PATCH → update part of the resource

2. Idempotent methods
Calling the API multiple times returns same result:
GET, PUT, DELETE.

3. Schema Validation
Used to validate response structure using

```
given()
.when().get(url)
.then().assertThat().body(matchesJsonSchemaInClasspath("schema.json"));
```

4. Throttling / Rate Limit Testing
I repeatedly hit API in a loop to verify:
✔ 429 Too Many Requests
✔ Rate-limit header values
✔ Retry-after behavior

5. Contract Testing
Deloitte uses Pact / Pactflow.
It ensures Provider & Consumer both follow agreed API structure.

6. API Authentication (Auth)
Deloitte often checks whether you understand how APIs stay secure.
They expect you to explain:

Common Types of API Authentication
API Key → Simple token passed in headers
Bearer Token → Encrypted access token
OAuth 2.0 → Widely used for user-based access (Google, Microsoft login)
JWT (JSON Web Token) → Signed token containing user info, expiry, scope
What They Want to Hear
How to pass tokens in Postman?
How token expiry is handled?
How you validate authenticated APIs?
Negative tests → expired token, invalid signature, missing authorization header

7.API Rate Limiting
Key Points
Rate limits protect APIs from overuse (e.g., 100 requests/minute)
Used to prevent DDoS, traffic spikes, bot attacks
Response codes:
429 – Too Many Requests
Retry-After headers

How to Answer
Explain how to test:
Burst traffic scenarios
Continual load over time
Validating correct 429 responses
Retrying after "Retry-After" cooldown

SECTION 4: DATABASE TESTING — Detailed Answers

1. Find 2nd highest salary without LIMIT
SELECT MAX(salary)
FROM employees
WHERE salary < (SELECT MAX(salary) FROM employees);
2. How to test stored procedures?
✔ Validate input/output parameters
✔ Validate business logic
✔ Test exception handling
✔ Compare output with expected DB state

3. Data Migration Testing
I compare:
✔ Count check
✔ Column-by-column validation
✔ Nullability/constraints
✔ Referential integrity

SECTION 5: MOBILE TESTING — Detailed Answers

1. Native vs Hybrid vs Web Apps
Native – built for OS (Android/iOS)
Hybrid – wrapped web views
Web – mobile browser-based

2. How do you test gestures?
Using Appium Touch Actions:
Swipe → scroll → tap → pinch → long press.

3. Common mobile defects
App crashes
Slow response
Memory leaks
Broken UI on different resolutions
Push notification issues

SECTION 6: WEB APPLICATION TESTING — Detailed Answers

1. How do you test a scalable web application?
I check:
✔ Load & stress handling
✔ Session management
✔ Caching behavior
✔ API response time
✔ DB connection pooling

✔ Failover & recovery under load

2. Explain cookies, sessions & tokens.
Cookies – stored in browser (client side)

Sessions – stored on server, mapped to a session ID

Tokens (JWT/OAuth) – encrypted authentication, stateless, modern apps use this

3. Accessibility Testing
I check:
✔ Screen reader compatibility
✔ Alt text for images
✔ Keyboard navigation
✔ Contrast ratio
✔ ARIA labels

Tools: AXE, Lighthouse.

4. CSRF & XSS Testing
XSS: Enter alert(1) in fields to check JS injection.

CSRF: Use modified tokens/invalid tokens to test unauthorized actions.

SECTION 7: AGILE — Detailed Answers

1. Explain Scrum ceremonies.
Daily Standup → 15 mins
Sprint Planning → capacity + story selection
Grooming → refine stories
Sprint Review → demo
Retro → discuss improvements

2. Handling last-minute changes
I evaluate:
✔ Impact on testing
✔ Scope change
✔ Effort
✔ Dependency

Then I update test plan + communicate risks early.

SECTION 8: HR + BEHAVIORAL — Detailed Answers

1. High-impact defect you found
"I identified a defect in payment rounding logic which caused incorrect tax calculation. Fixing it prevented financial losses during month-end billing."

2. Conflict resolution example
"A dev disagreed with a defect classification; I reproduced the issue with logs + video, and we mutually agreed on RCA."

3. Why Deloitte?
Strong QA practice
Opportunity to work on enterprise-level projects
Continuous learning culture
Global exposure

4. How do you handle pressure?
Prioritize tasks based on impact
Collaborate closely with dev/BA
Use checklists
Focus on quality, not just speed