**🔍 Terms Every QA Should Know (Explained at Low Level)**

**1. Bug / Defect / Issue**

* **Bug**: A flaw or fault in a software program that causes it to behave unexpectedly or incorrectly.
* **Example**: A login button does nothing when clicked.
* **Also Called**: Defect, Fault, or Issue – all mean the same thing in different contexts.

**2. Test Case**

* A **documented scenario** that includes steps to verify whether a software feature works correctly.
* **Includes**:
  + Test case ID
  + Test description
  + Pre-conditions
  + Test steps
  + Expected result
  + Actual result
* **Example**:
  + Test Case ID: TC\_001
  + Description: Verify login with valid credentials
  + Steps: Enter username → Enter password → Click login
  + Expected: User should be logged in

**3. Test Scenario**

* A **high-level situation** to test a specific function or feature.
* More **abstract** than test cases.
* **Example**: "Verify login functionality" – this could include many test cases.

**4. Test Plan**

* A **strategic document** that outlines the approach, scope, schedule, and resources for testing.
* Created before testing begins.
* **Includes**:
  + What to test
  + Who will test
  + What tools to use
  + Risk assessment

**5. Test Strategy**

* A **high-level document** that defines the testing approach for the entire project or organization.
* It’s **organization-wide**, unlike the test plan which is project-specific.

**6. Severity vs Priority**

* **Severity** = **Impact of the bug** on the system.
  + Example: "App crashes" = High severity.
* **Priority** = **How soon** the bug should be fixed.
  + Example: “Spelling mistake on homepage” may be low severity but **high priority** before release.

**7. Regression Testing**

* Re-testing the **existing features** to ensure new code changes haven't broken anything.
* Happens after bug fixes, code changes, or enhancements.

**8. Smoke Testing**

* A **quick basic test** to ensure that major functions of the software are working.
* Also called **Build Verification Testing**.
* Example: Open the app → Check login works → Check dashboard loads.

**9. Sanity Testing**

* After receiving a **minor change or bug fix**, test the specific functionality.
* It’s **narrow and deep**, whereas smoke testing is **broad and shallow**.

**10. White Box Testing**

* Testing the **internal structure or code**.
* Done by **developers** mostly.
* Example: Unit Testing.

**11. Black Box Testing**

* Testing without knowing the internal code.
* Focuses on **input → output**.
* Done by **QA testers**.

**12. System Testing**

* Testing the **entire application** as a whole after integration.
* Checks **end-to-end** flow.

**13. Integration Testing**

* Testing **interfaces between different modules**.
* Example: Login module → Dashboard → Profile → Logout.

**14. Unit Testing**

* Testing **individual functions or modules** of code in isolation.
* Performed by **developers**.

**15. UAT (User Acceptance Testing)**

* Testing done by **end users or clients** to ensure the product meets their requirements.
* Final stage before production.

**16. Automation Testing**

* Using **tools/scripts** to run tests automatically.
* Common Tools: Selenium, TestNG, JUnit, Cypress, etc.

**17. Manual Testing**

* Tester performs test **step-by-step manually**.
* Best for exploratory, usability, and ad-hoc testing.

**18. Test Data**

* The **input data** used during testing.
* Example: username = “testuser”, password = “Test@123”.

**19. Test Environment**

* The **setup where testing happens**: includes server, database, application, browser, etc.
* Should be similar to production.

**20. STLC (Software Testing Life Cycle)**

* The process or phases in testing:
  1. Requirement Analysis
  2. Test Planning
  3. Test Case Development
  4. Environment Setup
  5. Test Execution
  6. Test Closure

**🧠 As a Beginner QA, You Should:**

1. **Understand these terms thoroughly**
2. **Know the testing types (manual vs automation)**
3. **Write basic test cases**
4. **Know how to report bugs**
5. **Practice with tools like**:
   * **JIRA** (for bug tracking)
   * **TestLink / Xray** (for test management)
   * **Selenium** (for automation basics)

Would you like me to provide:

* A sample test case template?
* Real project scenario with test cases?
* Practice questions or interview Q&A?

**🔍 In-Depth QA Terms Explanation for Beginners**

**1. Bug / Defect / Issue**

* **Bug**: A mismatch between expected behavior and actual result in a software application.
* **Causes**:
  + Wrong logic by developer
  + Requirement misunderstanding
  + Incomplete coding
* **Example**:
  + User clicks "Submit" on a form → nothing happens (expected: it should submit).
* **Bug Lifecycle**:
  + New
  + Assigned
  + Open
  + Fixed
  + Retest
  + Closed or Reopened

**2. Test Case**

* A detailed **step-by-step** procedure used to test a specific feature or function.
* Includes:
  + **Test Case ID** – unique identifier
  + **Title** – short description of test
  + **Preconditions** – what needs to exist before test
  + **Test Steps** – clear instructions to perform
  + **Expected Result** – what should happen
  + **Actual Result** – what really happened
  + **Status** – Pass/Fail

**Example**:

|  |
| --- |
| Test Case ID: TC\_Login\_001  Title: Login with valid credentials  Precondition: User is on login page  Steps:  1. Enter valid username  2. Enter valid password  3. Click Login  Expected Result: User is navigated to dashboard  Actual Result: User navigated to dashboard  Status: PASS |

**3. Test Scenario**

* A **high-level idea** of what to test.
* It often covers multiple test cases.
* Helps with test planning and test coverage.
* **Example**:
  + Scenario: Test the shopping cart feature
  + Related Test Cases:
    - Add item to cart
    - Remove item
    - Checkout process
    - Apply coupon code

**4. Test Plan**

* A **master document** created by a test manager or lead to outline the entire testing effort.
* **Includes**:
  + Scope of testing
  + Test items (what will be tested)
  + Testing schedule
  + Roles and responsibilities
  + Entry & Exit Criteria
  + Risk Management

**5. Test Strategy**

* More **general and long-term** than the test plan.
* Describes the **overall testing approach**:
  + What testing levels (Unit, Integration, System)
  + What types (Manual, Automation)
  + Which tools (Selenium, JIRA, TestNG)
  + Communication and reporting methods

**6. Severity vs. Priority**

|  |  |  |  |
| --- | --- | --- | --- |
| **Term** | **Who Decides** | **Meaning** | **Example** |
| Severity | Tester | How bad the bug affects the system | Crash = High Severity |
| Priority | Developer/Manager | How soon the bug needs to be fixed | Login fails = High Priority |

**7. Regression Testing**

* Rechecking **existing features** after any code changes to ensure nothing broke.
* Example:
  + Developer fixed a bug in "search" → now QA retests login, cart, and checkout too.

**8. Smoke Testing**

* A **quick check** to ensure the application is **stable enough** to proceed with detailed testing.
* Often done after a new build is deployed.
* Example:
  + Can you open the application?
  + Can you login?
  + Can basic navigation work?

**9. Sanity Testing**

* Narrow and focused testing **after a fix** to verify that the specific problem was resolved.
* Faster and deeper than smoke testing.
* Example:
  + A bug was fixed in the cart → sanity test only the cart functionality.

**10. White Box Testing**

* Also called **Glass Box Testing**
* Tester knows the **internal code structure**.
* Performed by developers.
* Example: Testing a loop, if-else condition in the code.

**11. Black Box Testing**

* Tester does **not need to know code**.
* Focus on inputs and outputs.
* Performed by QA/testers.
* Example: Entering username/password and checking if login works.

**12. System Testing**

* Testing the **complete application** as a whole system.
* Performed in an environment similar to production.
* Example: Testing end-to-end flow – login → search → add to cart → pay → logout.

**13. Integration Testing**

* After unit testing individual modules, we test how they **work together**.
* Example:
  + Module 1: Login
  + Module 2: Dashboard
  + Test: Login → does dashboard load properly?

**14. Unit Testing**

* Performed by developers.
* Focused on **individual functions or methods**.
* Automated using tools like JUnit, TestNG.

**15. UAT (User Acceptance Testing)**

* Performed by **end-users or clients**.
* Confirms the software meets business needs.
* Final step before releasing the product to market.

**16. Automation Testing**

* Using **tools and scripts** to execute test cases automatically.
* Advantages:
  + Fast
  + Reusable
  + Saves time in regression testing
* Tools:
  + Selenium (web)
  + Appium (mobile)
  + JUnit/TestNG (Java test frameworks)

**17. Manual Testing**

* Tester performs **all steps manually**.
* Best for:
  + Exploratory testing
  + UI/UX testing
  + Short-term projects

**18. Test Data**

* **Sample or dummy data** used to execute test cases.
* Types:
  + Valid Data (correct input)
  + Invalid Data (wrong format)
  + Boundary Data (edge values)

**19. Test Environment**

* The **technical setup** where testers test the application.
* Includes:
  + Servers
  + Database
  + OS
  + Browsers
  + Network configurations

**20. STLC (Software Testing Life Cycle)**

Every testing project follows this life cycle:

1. **Requirement Analysis** – Understand what needs to be tested
2. **Test Planning** – Strategy, schedule, resources
3. **Test Case Development** – Write test cases
4. **Environment Setup** – Prepare testing environment
5. **Test Execution** – Run test cases
6. **Test Closure** – Final reporting, metrics, test summary

**🧪 Example: Real-Life Mini Test Flow**

**Project: E-Commerce Website**

**Scenario: Add Item to Cart**

|  |  |  |
| --- | --- | --- |
| **Step** | **Activity** | **Tool** |
| 1 | Understand the requirement (Add to Cart) | JIRA |
| 2 | Write Test Scenarios | Excel/TestLink |
| 3 | Write Test Cases | TestLink |
| 4 | Prepare Test Data (valid product) | Excel |
| 5 | Execute Test Cases manually | Browser |
| 6 | Log Bugs if any | JIRA |
| 7 | Perform Regression Testing after fixes | Selenium |

🚀 Tools You Should Learn as a Beginner QA:

|  |  |
| --- | --- |
| **Category** | **Tools** |
| Test Management | TestRail, Xray, TestLink |
| Bug Tracking | JIRA, Bugzilla, Azure DevOps |
| Automation (Web) | Selenium, TestNG, JUnit |
| API Testing | Postman, REST Assured |
| Performance Testing | JMeter |
| CI/CD | Jenkins, GitHub Actions |

If you'd like, I can provide:

* A sample **Excel format for test cases**
* Beginner **Selenium Java example**
* **Common QA interview questions** with answers