**Retesting :**

**Retesting** is a type of software testing that focuses on running the same test cases again after a bug has been fixed, to ensure that the issue has been resolved and the software now behaves as expected.

**Real-Time Analogies to Understand Retesting:**

1. **Fixing a Leaky Faucet**: Imagine you have a leaky faucet in your kitchen. You call a plumber to fix it, and after they repair it, you **turn the faucet on again** to check if the leak is gone. You are testing the same faucet, in the same way, to make sure the **repair worked** and there’s no more leak.
   * In this analogy, the **leaky faucet** represents the **bug** in the software, and **retesting** is like running the same test on the faucet after it has been fixed to ensure the problem is truly solved.
2. **Returning a Pair of Shoes**: You buy a pair of shoes, but when you try them on at home, you realize that the stitching is loose. You take them back to the store, and the store fixes the stitching. Once they give you the repaired shoes, you try them on again in the exact same way as before to confirm that the stitching is now fine and the shoes fit well.
   * This is similar to **retesting** a bug in the software. After the issue (loose stitching) is fixed, you test the software again in the same conditions to verify that the bug has been fixed and the functionality works as expected.
3. **Fixing a Flat Tire**: You have a flat tire on your bicycle. You take it to the mechanic, and after they replace the tire, you test it by riding your bicycle in the same way you did before the tire went flat. This is to make sure that the **flat tire** (the problem) is completely fixed and you can now ride without any issues.
   * **Retesting** in this case means testing the same functionality (riding the bicycle) after the fix has been applied (new tire), ensuring everything works as it should.

**Key Features of Retesting:**

1. **Same Test Case**: In **retesting**, you run the same test case that previously failed, after the bug has been fixed, to confirm that the issue has been resolved.
2. **Bug Fix Verification**: The primary goal of retesting is to ensure that a specific defect or bug has been successfully fixed and no longer occurs.
3. **Same Environment**: The tests are generally executed in the **same environment** (e.g., the same device, operating system, or browser) as the initial test to ensure consistent results.
4. **Focused Testing**: Retesting typically focuses on **one specific issue** that has been fixed, rather than testing the entire software.

**When is Retesting Done?**

* **After a Bug Fix**: When a developer fixes a bug, testers perform retesting to confirm that the fix works and the bug no longer appears.
* **After Code Changes**: When code changes are made to address an issue, retesting is done to verify that the specific change has resolved the issue without introducing new problems.

**Example of Retesting:**

Let’s say you’re testing an **online shopping website**.

1. **Bug**: A bug is reported where the "Add to Cart" button doesn’t work when clicked. Testers create a test case for the "Add to Cart" button, and it fails because clicking the button does nothing.
2. **Bug Fix**: The development team fixes the bug, and now the "Add to Cart" button works as expected.
3. **Retesting**: Testers **re-run the same test** (clicking the "Add to Cart" button) in the same conditions to verify that the button now works correctly and the issue has been fixed.

**Benefits of Retesting:**

* **Verification of Fixes**: Retesting ensures that the **problem** that caused the failure has been properly fixed and no longer exists in the software.
* **Prevents Reoccurrence of Bugs**: By retesting after fixes, you can ensure that previously fixed bugs don't show up again after changes.
* **Quality Assurance**: Retesting helps ensure that fixes have not introduced new problems or caused other features to break.

**Difference Between Retesting and Regression Testing:**

* **Retesting** is focused on **verifying a specific fix** for a particular issue, using the same test cases that previously failed.
* **Regression Testing**, on the other hand, involves testing the entire system or related parts to ensure that the changes haven’t affected anything else in the software.

**In Summary:**

**Retesting** is like checking if a problem has been fixed. After a bug is reported and fixed, testers rerun the same tests to confirm the issue is resolved. It's a focused process that ensures specific bugs are addressed, helping maintain the quality and reliability of the software.