Regression Testing

Regression testing is a type of software testing that ensures recent code changes have not introduced new defects or broken existing functionality. It is performed after modifications like bug fixes, enhancements, or configuration changes to verify that the application still works as expected.
Regression testing is crucial for maintaining software quality and stability after updates. While manual testing works for small projects, automated tools make regression testing faster and more efficient for larger applications.

Purpose of Regression Testing

- Prevents new changes from breaking existing functionality.
- Ensures software stability across updates.
- Helps detect unintended side effects of modifications.
- Saves time and cost by identifying bugs early.

When to Perform Regression Testing?

- **After Bug Fixes**: To ensure the issue is resolved without affecting other parts.
- **After New Feature Additions**: To confirm that the new feature does not break existing functionalities.
- **After Performance Improvements**: To verify that optimization changes do not introduce defects.
- **After Code Refactoring**: To make sure restructuring does not impact behavior.
- **After Integration with Other Components**: To check for compatibility and unexpected interactions.

Types of Regression Testing

1. **Unit Regression Testing**: Tests individual units/modules after small code changes.
2. **Partial Regression Testing**: Tests the modified module along with its dependent modules.
3. **Complete Regression Testing**: Tests the entire application to ensure overall stability.
4. **Selective Regression Testing**: Runs only specific test cases affected by the change, saving time.
5. **Progressive Regression Testing**: Applied when software requirements change, ensuring new updates work with old test cases.

Example: A banking application has a feature that allows users to transfer money. A developer updates the application to support international transactions.
Steps of Regression Testing:

1. Identify Test Cases for Regression Testing
   - Check if the new international transfer feature works.
   - Verify that existing domestic transfers are still functional.
   - Ensure login, account balance, and notification features remain unaffected.
2. Execute the Test Cases
   - Run automated and manual tests on old and new functionalities.
3. Analyze Results
   - If any existing feature fails after the change, report the defect.
4. Fix Issues & Retest
   - Developers resolve issues, and regression tests run again.
5. Final Confirmation
   - Once all tests pass, the new feature is deployed.

Regression Testing Strategies

- **Retest All**: Runs all test cases but is time-consuming.
- **Test Case Prioritization**: Executes high-priority test cases first.
- **Automated Regression Testing**: Uses tools to run tests quickly and efficiently.

Regression Testing Tools

- **Selenium** – Web application testing

- **JUnit, TestNG** – Java-based testing frameworks
- **Appium** – Mobile app testing
- **Jenkins** – Continuous integration and regression testing