

Software Regression Testing

In IT world a "regression" means **the return of a bug**.

Regression means errors that **occurs due to some action or activities** in a system.

Regression Testing

- *Regression testing is the re-execution of some subset of tests that have already been conducted to ensure that **changes have not propagated unintended side effects.***
- Whenever software is corrected, some aspect of the software configuration (the program, its documentation, or the data that support it) is changed.
 - Regression testing helps to ensure that changes (due to testing or for other reasons) **do not introduce unintended behavior or additional errors.**

Regression:
"when you fix one bug, you
introduce several newer bugs."



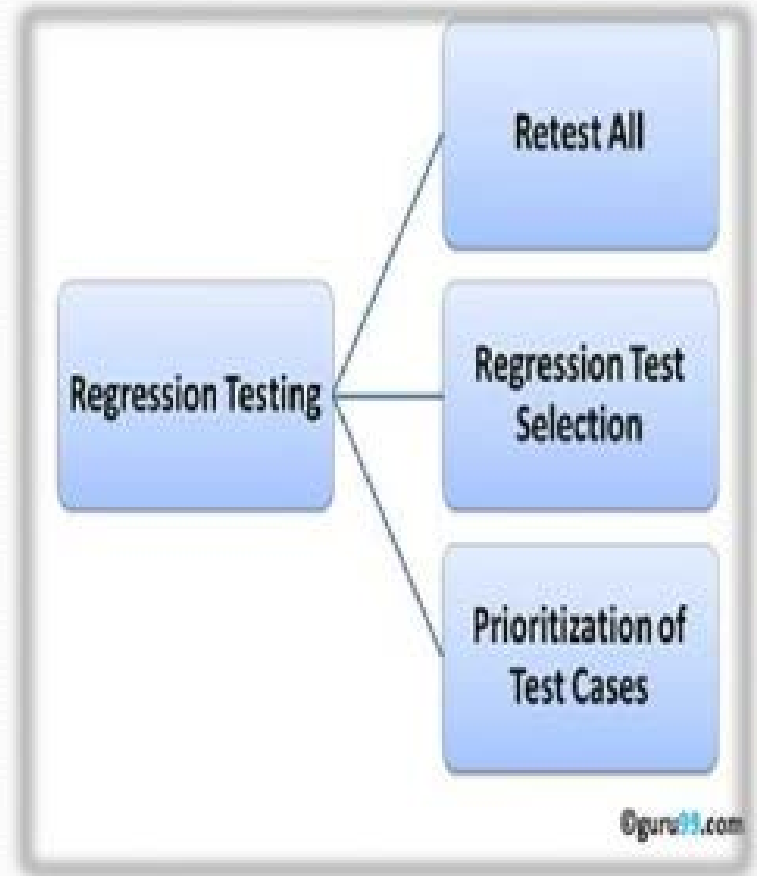
When to do it

Regression Testing is required when there is a :

- **Change in requirements and code is modified according to the requirement**
- **New feature is added to the software**
- **Defect fixing**
- **Performance issue fix**

What's the strategy?

- Software maintenance is an activity which includes enhancements, error corrections, optimization and deletion of existing features.
- These modifications may cause the system to work incorrectly.
- Therefore, Regression Testing becomes necessary.
- Regression Testing can be carried out using following techniques:



What's the strategy? (Cont...)

- **Retest all** - This is one of the methods for regression testing in which all the tests in the existing test bucket or suite should be re-executed.
- This is very expensive as it requires huge time and resources.

What's the strategy? (Cont...)

- **Regression Test Selection**
- Instead of re-executing the entire test suite, it is better to **select part of test suites.**
- Focus on software **functions that are likely to be affected by the change.**
- Focus on the **software components that have been changed.**

- **Prioritization of Test Cases**

- Prioritize the test cases depending on business impact, critical & frequently used functionalities.
- Selection of test cases based on priority will greatly reduce the regression test suite.

- **Selecting Regression Tests:**
- Requires knowledge about the system and how it affects by the existing functionalities.
- Tests are selected based on the area of frequent defects.
- Tests are selected to include the area, which has undergone code changes many a times.
- Tests are selected based on the criticality of the features.
- Test cases of Functionalities which has undergone more and recent changes.
- All Integration Test Cases.
- All Complex Test Cases.

Software Regression Process

Software Regression Process:

Step #1: Software Change Analysis

- Understand and analyze various software changes.

Step #2: Software Change Impact Analysis

- Understand and analyze software change impacts

Step #3: Define Regression Test Strategy and Criteria

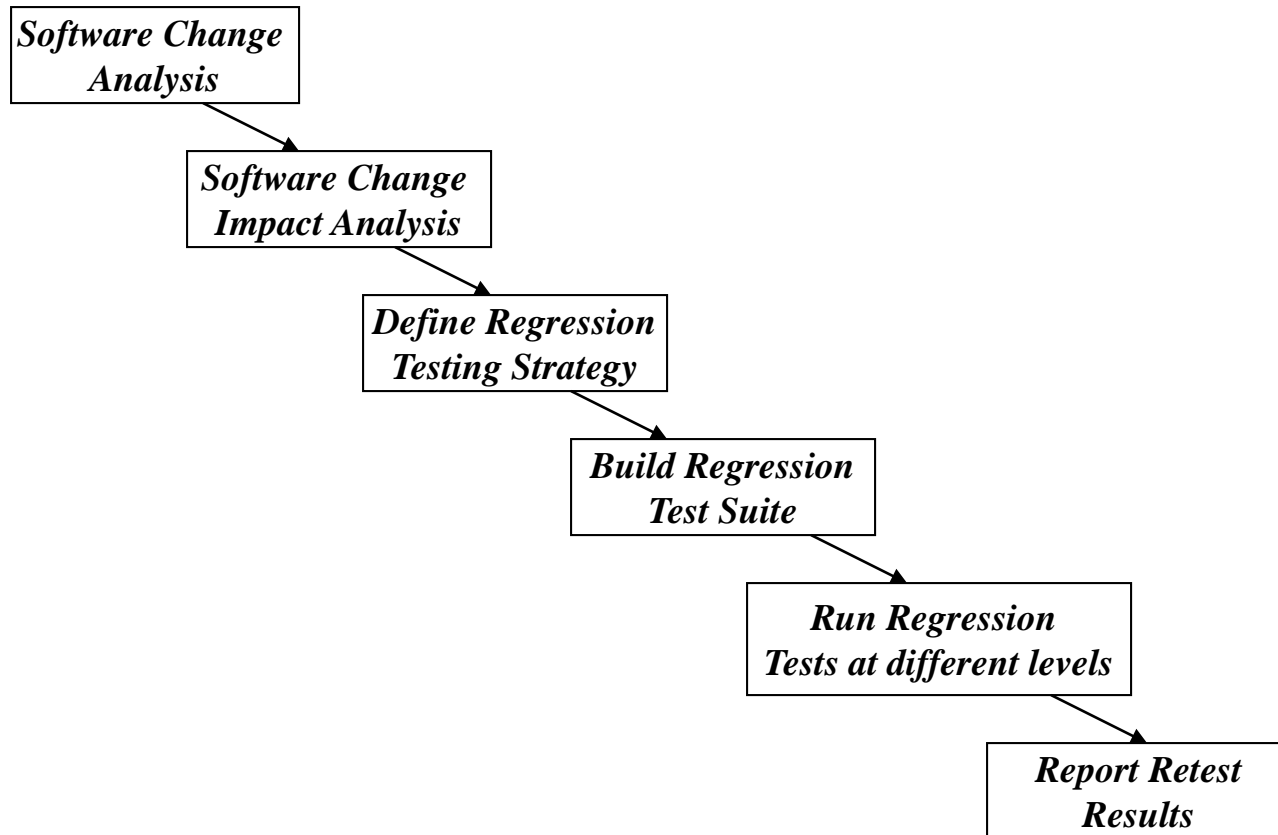
Step #4: Define, select, and reuse test cases to form a regression test suite

Step #5: Perform re-testing at the different levels.

- re-testing at the unit level
- re-testing at integration level
- re-testing at the function level
- re-testing at the system level

Step #6: Report and analyze regression test results

Software Regression Process



- Regression Testing ensure that the new changes hasn't affected other parts of the software.
- **Re-Testing** means testing the functionality or bug again to ensure the code is fixed.
- If it is not fixed, defect needs to be re-opened. If fixed, defect is closed.

Challenges for regression testing:

- With successive regression runs, test suites become fairly large.
- Due to time and budget constraints, the entire regression test suite cannot be executed.
- Minimizing test suite while achieving maximum test coverage remains a challenge.

Conclusion

- Regression Testing is the verification method.
- It verify that the bugs are fixed and the newly added features hasn't affect other parts of the software.
- Regression testing is very important that when there is continuous improvements added in the application.

Test Scenarios and Test Cases

- Prepare Test Scenarios
- Prepare Test Cases
- Arrange Test Data

- Example:
- If you have to withdraw money from an ATM machine, then it is a **scenario**.
- But to withdraw money, you need to execute many **test cases**, needs to provide.
-

- Checking the functionality of Login button is Test Scenario.
- Test Cases for this Test Scenario:
 1. Click the button without entering user name and password.
 2. Click the button only entering User name.
 3. Click the button while entering wrong user name and wrong password.

- Positive (regular, expected)
- Negative: Broken or missing data
- Exceptional: Check for the exception caught properly or not.