

Mutation Testing and Regression Testing

Software Engineering DA

Adhitya Nantish 21BDS0292

Software Testing

Brief intro

- Software testing is a testing which is conducted to provide information to client about the quality of the product under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.
- Software testing not only detect an error it operate the controlled condition of the software to (1) verify that it work correctly (2) to find an errors and (3) to validate the needs and requirements of user and the system work accordingly.
- In this presentation , we are going to focus on two main software testing - mutation testing and regression testing.

Software Testing

1. Verifications: the checking of the software consistency by checking the results against the requirements as already specified.
2. Error Detection: Testing should intentionally attempt to make things go wrong to determine if things happen when they shouldn't or things don't happen when they should.
3. Validation checks the system correctness –i.e. the process of checking and fulfilling the needs of the client.

Mutation Testing

- It is a process of testing in which code is modified then mutated code is tested against test suites. The mutations used in source code are planned to include in common programming errors. A good unit test typically detects the program mutations and fails automatically. Mutation testing is used on many different platforms, including Javascript, Java, C++, C# and Ruby.
- Mutation testing is a procedural testing method, i.e. we use the structure of the code to guide the test program, A mutation is a little change in a program. Such changes are applied to model low level defects that obtain in the process of coding systems
- Ideally mutations should model low-level defect creation

- Through mutation testing, we can find whether the set of testing methods used in the development process were good and correct to ensure final product quality
- Correctness of the testing schedule can be determined by checking the behaviour for each test conditions
- If there are no defects detected, then the program is appropriate or it passes the test
- For testing the program we take set of input values as a test condition so that it works according to needs and the corresponding output values according to the specifications
- Mutation testing is the test to check the consistency and accuracy of a testing program to find the bugs in the program by using different test conditions
- Mutation testing is a fault-based testing strategy because the faults are created by introducing a single error into the original program.
- There are many versions of mutation testing like weak mutation, interface mutation and specification based mutation
- The method was first introduced by Richard Lipton in 1971
- Mutation testing's working method is simple to use. In mutation testing we replace a logical operator with its inverse. For example, operator "!=" is used in place of "==" In some cases, mutation involves rearranging lines to change the execution order or even deleting a few lines of code
- If the program is modified, then test suite are executed against the code. The mutated code passes or fails the unit test, depending on the testing condition. Then a written unit test must detect mutated code errors, resulting in failure. A unit test that fails to detect code errors may require a rewrite.

Mutation testing basic examples

Pseudocode

The result of applying one mutation operator to the program is called a **mutant**. If the test suite is able to detect the change (i.e. one of the tests fails), then the mutant is said to be killed.

Let's take example to find a maximum between two numbers:

```
function MAX(M<N:INTEGER)
  return INTEGER is
begin
  if M>N then
    return M;
  else
    return N;
  end if;
end MAX;
```

Test Case

Take , $M=1$, $N=2$

1. The original function returns 2
2. Five mutants: replace ">" operator in if statements by (>,<,<=or=)
3. Executing each mutant:
4. Adding test data $M=2$, $N=1$ will eliminate the latter live mutant, but the former live mutant remains live because it is equivalent to the original function. No test data can eliminate it.

Mutants	Outputs	Comparison
if $M \geq N$ then	2	live
if $M < N$ then	1	dead
if $M \leq N$ then	1	dead
if $M = N$ then	2	live
if $M < > N$ then	1	dead

Advantages & Disadvantages

Advantages

1. Program code fault identification
2. Effective test case development
3. Detection of loopholes in test data
4. Improved software program quality
5. Elimination of code ambiguity

Disadvantages

1. Difficult implementation of complex mutations
2. Expensive and time-consuming
3. Requires skilled testers with programming knowledge

Mutation Operators

A variety of mutation operators were explored by researchers. Mutation operators examples used for testing :

1. Statement deletion.
2. Change each Boolean expression with true and false.
3. Change each arithmetic operator with any other + with *, - and /.
4. Change each Boolean relation with any other > with >=, == and <=.
5. Change each variable with another variable declared in the same part (variable types should be the same).

Regression Testing

- Regression testing is a type of software testing that seeks to uncover new software bugs, or regressions, in existing functional and non-functional areas of a system after changes such as enhancements, patches or configuration changes, have been made to them. When defects are found during testing, the defect got fixed and that part of the software started working as needed. But there may be a case that the defects that fixed have introduced or uncovered a different defect in the software
- The way to detect these unexpected bugs and to fix them used regression testing. The main focus of regression testing is to verify that changes in the software or program have not made any adverse side effects and that the software still meets its need. Regression tests are done when there are any changes made on software, because of modified functions.

- Regression means errors that occur due to some action or activities in a system. In IT sector a “regression” means the return of an error or bug. Regression testing is a type of testing that emphasis on retesting at every change is made. Traditional regression tests are often partially automated. The numbers of regression testing have been developed for the optimization. The studies of the techniques suggest that regression testing can provide benefit to testers. Regression testing attempts to mitigate two risks:
 1. Changes that was planned to fix an error failed.
 2. Sometime modification made some problems like introducing a new bug

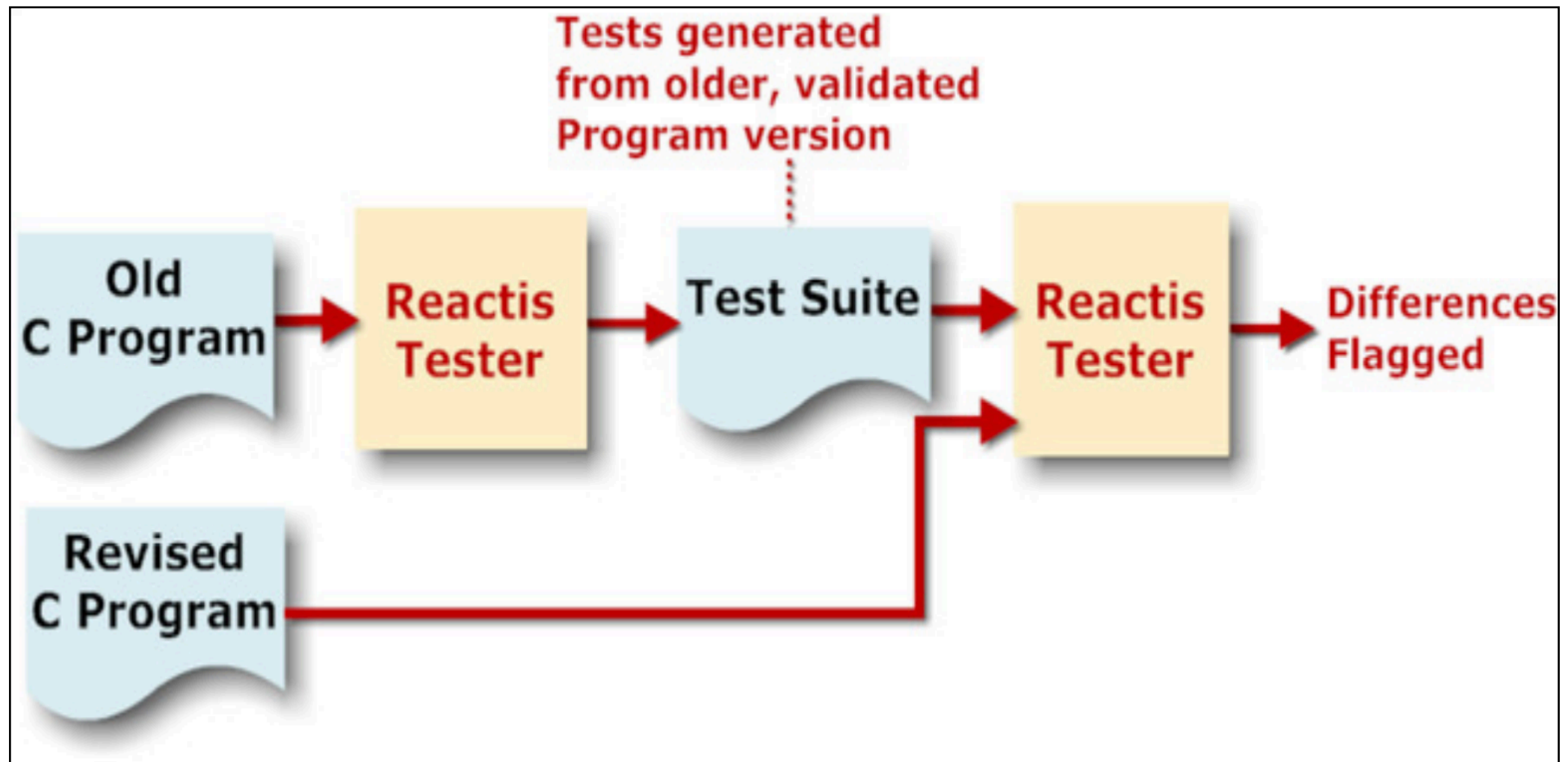
- Regression testing checks if old functions of an application are working correctly after modifications.
- It ensures that modifications haven't introduced new bugs.
- This test is essential when there are significant changes in original functions or bug fixes.
- Regression is a verification process to confirm that errors are fixed.
- It ensures that newly added functions don't cause issues in previously working software.
- Functional testing is performed when new parts of the program are available for verification.
- The focus is on verifying changes made to existing functions and new additions.
- After completing tests, software testers verify existing functions to ensure they work as expected.
- Regression testing is usually done after changes or the addition of new functionality.
- The process of releasing regression tests can take months, so it should be integrated into daily test cycles.
- Regression testing is applied when errors are fixed or new code is added to the software.
- It integrates new code with previous code to maintain software quality without affecting existing functionality.
- Testing teams also check last-minute changes in the system.
- In such cases, testing focuses on affected application areas to complete the process on time, considering major system changes and aspects.

Regression Testing Examples

Example of regression testing with its process is explained below:

Let's take example of college in which there are three main parts named Administration part, Personal Information, and Faculty. Suppose error occurs in the Administration part, existing User is not able to login with valid login details then error get occurred. Then Testing team pass the bugs to the Development team to correct it and when development team fixes the Bug and again give it to testing team than they checks that bugs get fixed and does not change the remaining functions of other parts and also the functions of the same part. This process is known as regression testing done by Software Testers.

1. Regression testing is required when there is an error in program.
2. Change in requirements and code is modified.
3. New feature is added to the software.
4. Bugs get removed and Defect gets fixed.



Regression Testing process