# A Study of Effective and Efficient Regression Testing on Object Oriented Software

Swarna Lata Pati

*Department of Computer Sc & Engg*
*College of Engg. & Tech., Bhubaneswar*

*Abstract* **-This paper presents a comparative study of effective and efficient regression testing on Object Oriented software . Regression testing refers to retest the unchanged parts of the application software Test cases are re executed to verify old version of application is running fine and new changes ( addition, deletion modification ) have not introduced any new bugs. Regression test is performed by running many or all the test cases created to test modifications in previous version of the software . Many researchers have contributed a lots in the area of regression testing, i.e. how to select regression tests so that the number of test cases does not grow too large as software evolves. Regression test case prioritization etc. To elucidate the effectiveness and efficiency to some more extent We have conducted some studies. Our proposed blended technique helps to conduct regression testing by minimal test case selection which reveals fault earlier , reduction of test-suite and prioritization of test suite. Researchers have contributed many angles for test case prioritization bearing on the use of coverage information . In my study I have taken JCOV code coverage tool to estimate coverage information as I propose to use JAVA as object as my programming language as it is purely object oriented , platform independent and an open source software. In our study we report our outcomes with the help of automated testing tools selenium, QTP . We have also used white box testing approach using JUNIT framework to test new units added or modified to the old version. Some empirical studies have been conducted by taking small subjects with small test suites into considerations. We share our experience with a step by step regression testing process.**

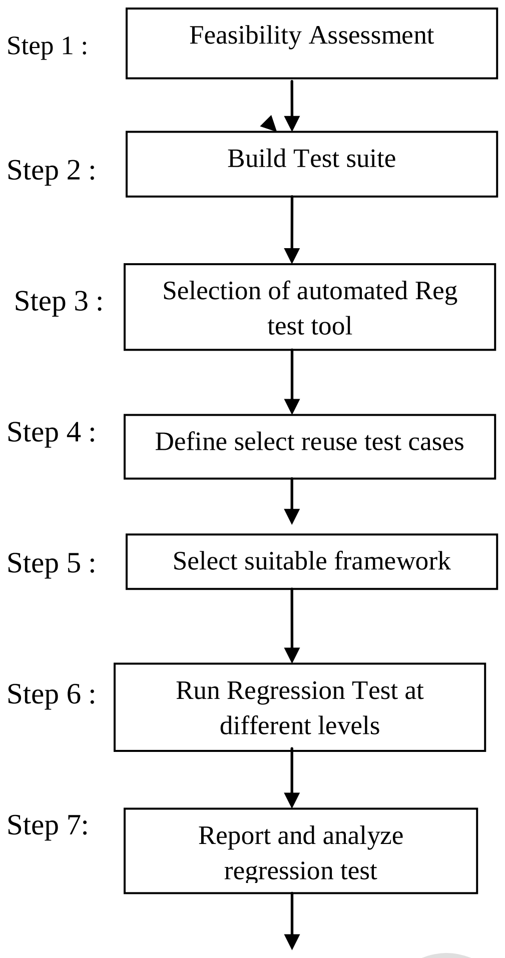
***Key words :Regression testing, software maintenance ,test case selection test minimization, automated test tools.***

## I. INTRODUCTION

Regression testing is an major retesting activity for software validation and maintenance. Test cases are re executed in order to check if previous functionality of the system is working fine and new changes have not introduced any new bugs. This test is very important when there is continuous change / improvements added in the application. The new functionality should not negatively affect existing test unit. Why regression tes? How much regression?what to do in regression? Are the main issues here . Regression testing is initiated when programmer fix any bug or new unit is added to the system. How much regression depends upon the scope of the newly added unit but for effective testing regression test plan and strategy would solve the problem to a larger extent. For software validation fewer test cases or arbitrarily omission of test cases use in regression testing is risky and produce less predictive unstable product. In this paper we investigate methods to select small subsets of effective fault revealing regression test cases to revalidate the software . Using JUNIT we can incrementally and cheaply build and select test suite which helps us to measure the progress notify unintended side effects and reduce effort. The fault set was obtained from error logs maintained during unit testing and integration phase. Many techniques have been reported in the literature [ 1 ],[8],[3] on how to select regression test, test prioritization . But the research questions here are How to identify changed software parts and its impact ?How , when and where to use to use regression tools ? The major challenges ahead are reduction of re-test suite along with effort minimization and achieve the adequate testing coverage ?To meet the requirements of specification changes our study presents a Regression test process model which is a blended one in section II.

## II. REGRESSION PROCESS MODEL

Regression testing process starts during software maintenance phase .For extensive low level interface manual testing is time consuming and laborious . Once automated regression tests cases have been developed they can be run quickly and repeatedly for the long maintenance life of software product The major testing objectives are –Retest changed components or parts - check changed components or units - check the affected parts or components for which we have to conduct regression test at unit level, reintegration level functional level at last at system level . To achieve software regression testing problem and to meet the challenges we propose a process model called automated process model for regression test . The propose medel goes through varios phases /steps . The steps are :


Step 1 :
Feasibility Assessment
Step 2 :
Build Test suite
Step 3 :
Selection of automated Reg test tool
Step 4 :
Define select reuse test cases
Step 5 :
Select suitable framework
Step 6 :
Run Regression Test at different levels
Step 7:
Report and analyze regression test

This model suggest the blending of manual with automated tools to produce the output as expected. Step-1 suggests prior to automate we conduct a feasibility study to check if the application is fit for automatic regression It depends upon test suite size, nature of application .Step-2 Prepare test suite as per the modified version of the software since the test suite for earlier version already available with the developer/tester . Step-3 Choose a particular regression test tool (automated) depending upon the language in which the test script/suite is written . step-4 Select , reuse test/suite from enumerated numbers of test cases/suite, we must prioritize [8] the test cases ,Prioritization is made as per code coverage criteria and fault revealing [2] suite . In our model we have again used Jcov code coverage tool to test the coverage information. Table1 of section III shows the details of coverage information. Step-5 Select suitable framework for testing depending upon the language chosen. In our research we have chosen JUNIT for java language as our framework. Here to ensure which major functionalities are to be automated. Framework is very crucial for the success of any automation.Step-6 Execute regression test at different levels by name of the function. Defects if any are logged in a log file Test cases are usually version specific. At last a comparision is made between expected and actual output from the unit. Actual results shows passed/failed. If pass – what actually happens when you run the test. If failed – put in description of what you have observed is the activities of step-7.d.

## III. SOFTWARE REGRESSION STRATEGY

Software regression strategy refers to a rational way to define regression testing scope, coverage, retesting sequence and reintegration orders. Code coverage process measures the code quality. In our study we have incorporated JCOV to measure code coverage . It stores all the test coverage information on open source database Oracle. A tool presented by [4 ] but JCOV supports both static instrumentation and dynamic instrumentation for coverage analysis. The data can be collected on the file system or onto a server. The JCOV tool is augmented with Java testing frameworks JUNIT to test the units of Java. The structure of the coverage measurement with JUNIT as tool. Prior to use JUNIT write your own source program.

```
.src/

.project.java // source program

.JUNIT

>projectname.java

.bin/

.lib/

.JUNIT.jar

.JUNIT_Test Runner.jar
```

JCOV can be used as a command line interface. For each coverage type probes are inserted at method, block branch and line. The probe is used to track coverage information. The probes are in form of identifier i.e. class-id, method-id, block-id and branch-id helps to measure coverage information after execution of test.The coverage–log database will store coverage data. Table I shows Coverage Tool comparision. Table II shows Coverage information for ChatClient.java program which is one the subjects of my experiment. Which is a client login in client-server application. In this study two versions of subjects are taken for ChatClient.java with 18 and 23 test cases respectively.

TABLE I
Coverage tool comparision

| Cov tool | Cov Type | Instrumentation | Requirements |
|---|---|---|---|
| Emma | Open | Byte code | No ext lib |
| Java codecoverage | open | Byte code | Mysql |
| JCov | open | Byte code | Oracle jdk |
| TACT | Prop | Byte code | NA |
| Jcoverage | both | Bytecode | NA |

TABLE II
Code coverage information for version 2 of ChatClient.java

| **Coverage type** | **No of units** | **No of coverage units** | **Coverage information %** |
|---|---|---|---|
| **Method** | **7** | **7** | **100** |
| **Branch** | **11** | **8** | **74** |
| **Line** | **51** | **41** | **80** |
| **Block** | **4** | **3** | **75** |

The coverage information can be extended further for relational operator, bit-wise operators but it is beyond the scope of this study .

Evaluation of prioritized test cases- Test case prioritization is an approach for sequencing the test cases in terms of fault detection capabilities as per G Rothermal et al [1 ]. Prioritise is to order the tests in a test-suite so that faults can be revealed as early as possible during testing. The test cases that are more likely to reveals fault should be run before test cases that are less likely to reveal faults . But in any case all the test cases are to run. Ordering can be performed by taking percentage of coverage information and percentage of of faults detected value. To model fault detection let us assume that p is the program for which a test suite Pt, t1,t2,t3 are set of test cases which reveals faults as f1,f2,f3,f4 .If we apply test case t1 it expose 3 faults ,on apply t2 no faults and on apply t3 , AUT(application under test) reveals two faults . In that case prioritization sequence is (t1,t3,t2). Even if t2 does not reveal any fault it is to be tested with low priority for longer maintenance life of the application.

## IV. FRAMEWORK SELECTION

The regression test process model selects JUNIT[10]as unit test tool as Java programming language is chosen for experiment subjects. It helps in development of Test driven development framework. JUNIT linked at JAR at compile time and used to write repeatable test. Using JUNIT we can incrementally build a test-suite that helps us to work with JUNIT tool. Prior to use JUNIT write your own test , when we would encounter new bugs catch them with test, fix them, observe again fix them and so on. Let us take one example to test a test case by using code a little test a little method. An easy method to write test suite using JUNIT.A test-suite is the group of unit test cases ready for running in JUNIT @RunWith and @suite annotations are used to run the test suite. Let us write two classes for two testcases as TstOne and TstTwo .

```
Test case #1

import org.junit.Test;

Public class TstOne{

   @Test

    Public void testcaseOne(){

 System.out.println(“Hi !I am in test one”);

          }

}
```

```
Test case #2

Import org.juint.Test;

Public class TstTwo(){

  @Test

Public void testcaseTwo(){

System.out.println(“Hi! I am in test  case two);

        }

}
```

For Test Suite create a java class called “TestSuite”augmented with @RunWith(Suite.class) plus @Suite.SuiteClasses.

The code should be like :

```
Import org.junit.runner.RunWith;

Import org.junit.runners.Suite;

@RunWith(Suite.class)

@Suite.Suiteclasses({

      TstOne.class, TstTwo.class})

Public class TestSuite{

}
```

Run the Test Suite classes with RUN.

JUNIT is GUI tool so by clicking on R**U**N .

Results show the output with failures and Errors with duration of running time. Advantages of JNUIT tool is that –GUI based Automatic execution of regression test allows us to add or update the test cases easily – for which regression tests must be incorporated in the daily test cycle -test report generation .Test Reports format :

Assertion Test name : MathTests:: validClassification

Equally assertion failed

-Expected : 1

- Actual : 4

Failures 3

Run: 4 Failure : 1 Failures:1 Errors : 0

Failed tests

Table III
Test report

Id name failure type location Message

1 Triangle.java Test assertion lineno12 F

Actual : 4 Expected : 1

If there is a change to any test case only test case file needs to be updated the driver script and start up script will remain same.

## V. AUTOMATED REGRESSION TESTING TOOL

In our study We have used automated regression testing tools to automatically test key functionalities tests for regression as well as run a large number of tests in very short period . Automation of the application allows you to verify web based interfaces, make changes and refactor code quickly after every change.There exits a good numbers of testing tools available in the open source[5]. Selenium, Waiter, Windmill, Sahi, QTP (open source version available for limited time and limited applications)[6[.Unit testing framework such as NUnit, CUnit, JUnit. We have already chosen JUnit for our study. Open source provides a competitive features for automating the testing of software applications. For functionality testing and regression testing of software we have chosen SELENIUM and QTP. Technical features of Selenium and Qtp [7] are given here-in-under for better understanding.

Features of Selenium :

1.GUI based

2. Open source

3.also used for unit testing

4.support all platform

5.java can be used to write test suite

6.Test is written through IDE

7.Uses command open,click,debug and execute

8.Easy record and play back

9.Auto complete walk through test

Selenium demo test report example given for subject Chatclient.java here in under in table IV.

Table IV
Selenium demo test report

Test suite : chatclient Tests

Source /home/workspace/seleniumlib/demo/login_tests

Overal status :Fail

Message: 10 critical tests 4 passed 6 failed

Time/start/end/elapsed :05:20:20 05:21:27

Test description Username empty,username wrong,username out of range , timeout,Similarly password wrong , password empty, time out any combination is possible for test case.

Features of QTP :

1.Facilitate automatic regression testing for softwar application.

2.QTP comes with a user interface that can be considered as an IDE for the test.

3.Supports java language UFT 11.5)

4.Runs only on Windows environment

5 Record and plaback test creation method

6 supports manual script writing using excel

7 License cost is high

8. Execution time is higher but Scripting time is less

9. Easy to use validate results and generate reports and supported by HP.

Here we would like to introduce the cost effectiveness of the regression test by introducing a cost model of a regression software here as : REg cost = Ana cost + EV(cost of executing ,validating all tests in modified software).

Analysis cost involves cost of development effort, time and maintenance effort. It also includes human effort in Person Month and equipment cost. By analyzing the cost incurred we propose to go for semi-automation in some case complete automation depending upon the size and complexity of regression.

## VI. EMPIRICAL EVALUATION

To evaluate my approach for blended technique I have conducted empirical studies by taking five subjects , test cases are generated in some case using Java in some case using excel and in some case using JUNIT framework results are presented in a table here-in-under. Each software subject consists of an original version P, several modified versions(v1v2 .. vn) and a set of test cases that was used to test P, Table shows the programs and for each subject list of methods in original program number of versions,size of test suite and percentage of methods covered by test suite .

Table V
Software programs used for Experiment

| Subject | Method | version | Test cases | Cov Inf% |
|---|---|---|---|---|
| Mean.java | | 3 | 11 | 80 |
| Gcd.java | | 2 | 19 | 82 |
| Xypow.java | | 4 | 16 | 85 |
| Triangjava | | 7 | 11 | 75 |
| Chatclient.java | | 3 | 7 | 90 |

(Software programs used for experimental study)
As the automated tools are open source, takes lesser time and thus the Cost of regression reduces.

## VII. CONCLUSIONS

In this paper, I have presented the blended process model for automated regression test tool through which a software undergoes regression test.I have presented regression strategy, apply of JUNIT with example and test results. I have also augment JCOV with JUNIT to determine code coverage information. This paper also presents relation between fault occurrence with test case. My study also indicates use of regression test tools available in open source with some results to meet the challenges of regression test and efficient regression can be achieved. My study indicates how to achieve cost and time effectiveness by using the appropriate open source tools.

Our future work will include regression test tools application to large test suite in case study based approach sucf that more effectiveness and efficiency can be achieved.

## REFERENCES

[1]. G Rothermel , M. J. Harrold, J Ostrin and C Hong .An empirical study of effects of minimization on the fault detection capabilities of test suiteI nIEEE computer society washington .

[2]. W Eric Wong Hira agrawal et al Moristown NJ."A study of Effective regression testing in practice ".

[3]. Regression Test selection for JAVA software J Harrold, james a Jones et alProc. Of ACM on OO Prog system .

[4]. A Multipurpose code coverage Tool for JAVA R Lingampally,A. Gupta, P Jalote Proceedings of Hawai Int Conf.on system Science 2007

[5]. www.testingtools.com

[6]. www.tutorialspoint.com

[7]. http://junit.sourceforge.net

[8]. Search Algorithms for Regression Test case Prioritization by Zheng Li. Mark Harman and Robert Hierone IEEE transactions of software engg.

[9]. Regression test selection for C++ software G Rothermel Marry J Harrold Jeinay Dedhia.

[10]. http://www.xoriant.com