

Automated Regression Test Model with Cost Estimation

Swarna Lata Pati * Manojranjan Nayak**

*College of Engg & Tech , BBSR, Bhubaneswar, Orissa

** SOA University , BBSR, Bhubaneswar, Orissa

Abstract: - This paper presents a Model to conduct automated regression testing. The model is evaluated for Java software. The efficiency of the model is evaluated from code coverage information. The effectiveness is measured by taking cost model into account. A cost model for automated regression test model is suggested by taking different cost component into considerations. In software industry , test automation is the use of special software, different from software being tested to control the execution of tests and comparison of actual outcomes with predicted outcomes. Test automation can automate some repetitive but necessary tasks in a formalized testing manner or add additional testing the would be difficult to conduct manually.

Keywords: *Regression testing, cost model, QTP, Test driven development, QA ,QTP.*

I. INTRODUCTION

Why to automate Some software testing task, like low level interface regression testing is laborious and time consuming. Irrespective of labor manual approach might not always be effective in finding certain classes of defects[1]. So our purpose is to develop automated regression test model so that tests can be automated quickly and repeatedly[2]. The objective of this model is to produce cost-effective method for regression testing of software, particularly for the product which is having a large maintenance life. Regression tests which are repetitive nature are given prior automation reduce effort and cost. Clients face multifaceted challenges in regression testing due to the constant stream of upgrades. The proposed model reduces regression testing effort . which later on reduces cost and produce effectiveness and also minimizes business interrupts. Different testing tools helps us to automate data setup generation ,product installation GUI interaction, defect logging . Out of numerous testing approaches the following approaches are used by researcher – Code driven testing – API driven testing – GUI interface testing. To get cost-effective in test automation in long term , especially used repeatedly in regression test. The Organization of the paper is presented as : Section II presents proposed model, section III presents control flow of regression test automation, section IV presents preautomation phase section V presents framework of the proposed model, Section VI presents

Cost model, Section VII presents Empirical evaluation and section VIII presents conclusion.

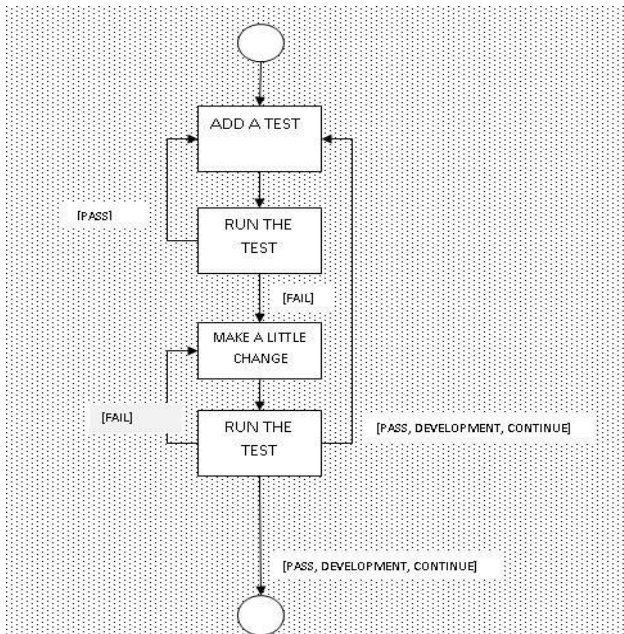
II. PROPOSED MODEL

As the business needs grow, the pressure to deliver the product in time at the same time to ensure quality, Parameters to be considered for automating software application are as follows – Data Driven capabilities,- debugging and logging capabilities, platform independence , - extensibility , - customizability, version control friendly and support unattended test runs. We face the challenges of improper undocumented test case, no standard format to store test case, proper methodology to select test cases etc . The model integrate the test management tools to improve scheduling capabilities to run test scripts in various environment.

III. CONTROL FLOW OF REGRESSION TEST AUTOMATION

There are basically four test automation frameworks that are adopted while automating the applications - Data driven automation framework – Keyword driven automation framework- Modular automation framework – Hybrid automation framework. Testing methodology can be adopted either of the following – Model based testing – Behaviour driven testing and code driven same as test driven testing (TDD[4]) . The step by step procedure for automation of QTP are as follows: test and debug the application – add something to the test application, Design a test feature added to the new build, - run both the old and new tests over the new build. These tests are added to the test sequence of the project. Fix and rerun until everything is clean. The flow diagram in TDD approach is given in diagram 1 .

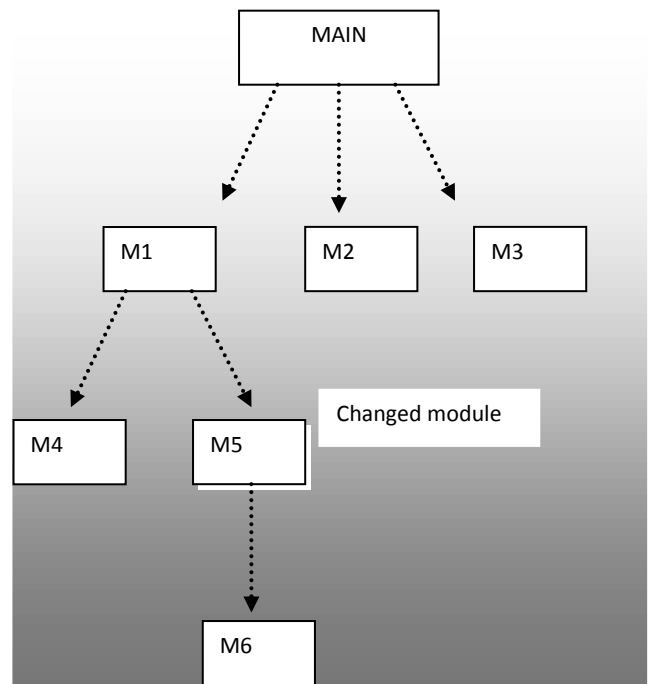
Diagram -1 [FLOW DIAGRAM OF ADDING A TEST AND RUNNING]



IV. PREAUTOMATION PHASE

The preautomation phase include the set of activities as – Gathering and understanding the existing regression test cases. – Documentation of core test cases in standard format in accordance with automation requirements. Choice of a particular methodology for test case selection , prioritization, minimization depending upon a particular criteria .Analyze the documented test cases for complexity, automation ability and scope for improvement. Categorization of test cases based on the respective operating unit the module and tracking mechanism. Unit tests are written to define the functionality before the code is written , however these unit tests evolve and extended as coding progresses. Issues are discovered and code is subject to refactor. So the group of test cases which pass is the complete code. It is more reliable and less costly than code that is tested by manual exploration. Code coverage is better that we will evaluate through empirical research. The developer discovers defects immediately , upon making a change , when it is least expensive to fix, code factoring is safer., transforming the code into simple form with less code duplication . So probability of introduction of new defects reduces. This enables the tester to create test cases by editing the parameter and test conditions. An example with a changed module for regression testing is cited for better understanding .

Based on the changed information we can select define and reuse class test for regression testing. Change may affect M5, M1 and main module. Retest at unit level i.e. Module M5 and reintegration with module 2, 3, 4 . Strategy used to identify changed classes are – identify affected classes , find out the order – apply class retesting at unit level [3]– use the class test order to integrate classes together. The diagrammatic representation of the modular integration is given below.:-



V. FRAMEWORK OF PROPOSED REGRESSION TEST MODEL

A regression test automation model framework is an integrated system that sets the rules of automation of a specific software product. This system integrates the function libraries, test data sources, object repository, object details, and various reusable modules as components. To meet the business need i.e.to develop a reusable and user friendly test automation framework and business logic is developed. Each component act as small building blocks which need to be assembled to represent a business process. This framework provides the basis of test automation and simplifies the automation effort, low cost of maintenance[5,6,10,11]. If there is any change to any test case then only the test case file needs to be updated and driver script and start up script will remain same hence reduces no of stubs in testing process. Choosing the right framework or scripting technique helps in maintaining lower cost. The cost associated with test scripting are due to development and maintenance effort. The approach of scripting used during test automation has effect on costs. The testing framework is responsible for defining the format in which to express expectation. It creates a mechanism to hook into or drive the application under test. The interface engine performs the work of parsing and executing and produce the reporting results. Regression test automation interfaces are platforms that provide a single workspace incorporating multiple testing tools and framework for system integration testing of AUT (application under test).The goal of test automation interface to simplify the process of mapping from test to business logic. The modules of regression test automation model consists of Interface Engine , Interface Environment, Object Repository . Regression test has been performed by QA team after development team has complete the work. Defects found at this stage are most costly to fix. This problem is being addressed by rise of

developers testing. The test is conducted by using QTP, an environment for development (IDE) i.e Eclipse, and execution through a client server architecture or simply through terminal server. How a regression test is performed in automated QTP is elucidated here in under : Design a data driven modularized automation model. Include parameterization , inclusion of library and utility functions . automating the identified test cases using QTP. Uploading test cases and regression scripts to the interface engine for effective management and administration.

The diagrammatic representation of model is given below :

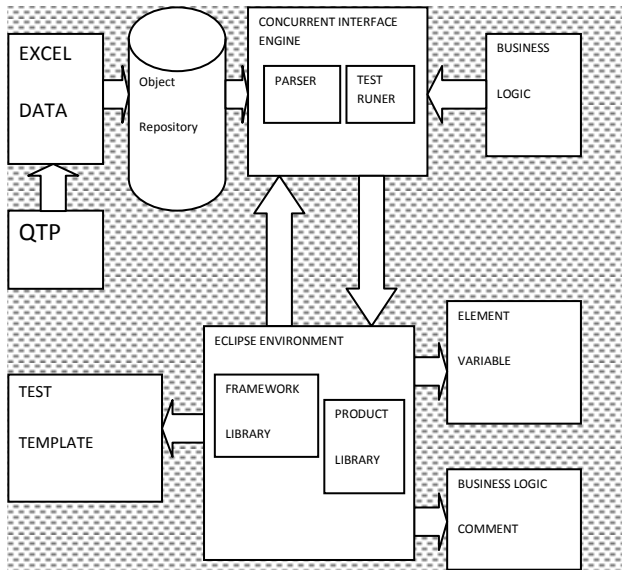


Diagram-3 [BLOCK DIAGRAM OF AUTOMATED REGRESSION TEST MPDEL]

VI. COST ESTIMATION MODEL

Cost model related to the automated regression test is designed .which is named as Cost Estimation model[7,8,9]. **AS Time** is the key of every regression testing. To model the cost of regression testing we require certain number of input to the cost model. The obsolete test cases do not add any cost. Out put of the model is product with a cost , it is the sum of all costing parameters.

Assumption for cost model: All test cases have uniform cost.

The cost factor is directly proportional to the no of faults detected. If the risk factor is taken in to account the risk or threat to failure is also responsible for inclusion of cost . So the cost of the regression test includes cost of resources associated with it . The input parameters to the cost model are given here in under: 1. Acquisition cost : It is the cost involved in the procurement process of the tool. It includes cost of the tool, cost based on no. of licenses and version upgrade cost. 2.Training cost : cost incurred with training imparted to the personnel. 3. Environment cost (Cost involved in setting up the environment) i.e. Cost of hardware and software. 4.. Development cost : Cost involved in development of framework. 5.. Maintenance

cost . The diagrammatic representation of cost Estimation model is presented here[diagram-4]. The associated cost equation computes the cost of regression tentatively.

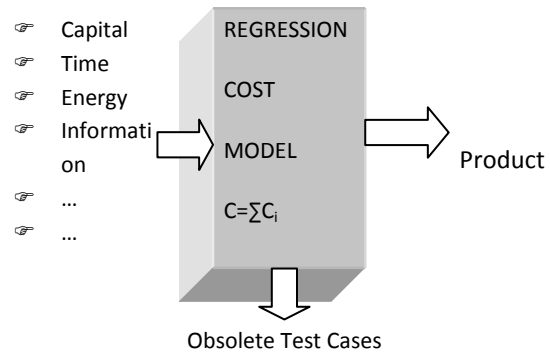


Diagram-4[COST ESTIMATION MODEL FOR AUTOMATED REGRESSION TEST MODEL]

Each product has its own maintenance cost. Taking above cost drivers into account. The output of the cost model is the sum of all the components with some adjustment factor.

Let C_x = cost of Analysis of regression test.

C_y = Cost of using a particular test strategy., N = number of test cases in that strategy .

C_R = number of regression test case selected.

$$(1-nf/nfo)X C_R + C_x + NC_Y$$

nf = number of faults , nfo = number of faults detected by original test suite.

C_E = cost of execution it includes man hour .

C_T = cost of validation all test includes time and effort.

Risk exposure cost $C_{Re} = p(f) X c(f)$

$P(f)$ = probability of occurrence of a fault

$C(f)$ = cost of impact of a fault in the corresponding function if it occurs in operation

Over head cost = C_{oh} / n

n is the number of tests pass per hour.

$$So \sum C_i = C_x + (1-nf/nfo)X C_R + NC_Y + C_{Re} + C_{oh}/n + C_T + C_E$$

VII. EMPIRICAL STUDIES AND EVALUATION

To evaluate our model we have conducted empirical studies The Java Application server was taken as a subject It is observed that the cost and effort of running each regression test cycle is reduced by 70-80% as compared to blended l one. Code coverage is more or less same .

VIII. CONCLUSION

While implementing regression test automation it is observed that automation reduces cost of saving, promoted software quality. Automating test and other repetitive tasks reduces time spent and enhance efficiency. Test automation requires less man hour and less person month thus reduces cost . Time is the key to any project , so I propose the researcher , students, industry to go for semi automation and automation if feasibility permits.

REFERENCES

- [1]. Regression test selection for C++ software G Rothermel Marry J Harrold Jeina Dedhia.
- [2]. <http://www.xoriant.com>
- [3]. Prioritizing JUnit test cases G Rothermel, A Kinneer ,Hyunsook Do
- [4]. Evaluating the efficacy of Test Driven development bT. Bhat and N Nagappan.
- [5]. Kaner Cem “ Architecture of Test automation” August 2000.
- [6]. Elfriede Dustin et al. “ Implementing automated software Testing “ Addison Welley ISBN 0-291
- [7]. Software Engineering a Practioner’s Approach. Roger S Pressman, Tata McGraw Hill.
- [8]. Fundamental of Software Engineering by Ian Sommerville. Pearson Edition.
- [9]. Fundamentals of Software Engineering by Rajib Mall, PHI edition.
- [10]. Bas Vodde and Lasse Koskela , IEEE Software Vol 24 issue 3, 2007
- [11]. Mark Fewster and Dorothy Graham (1999)” Software Test Automation “ ACM press ISBN 978-0-6-23372-7.