Amisha, Dr. Vinay Hegde

(Department of Computer Science and Engineering R.V. College of Engineering , Bengaluru , India amisha.cs17@rvce.edu.in )

(Department of Computer Science and Engineering R.V. College of Engineering Bengaluru , India vinayvhegde@rvce.edu.in )

----------------------------------------*************************-------------------------------

# A Brief Introduction on Automated Regression Detection Framework

## Abstract:

Detecting performance anomalies/defects and finding their root causes are tedious tasks which requires much manual work. Conventional approaches are limited to specific applications. This paper aims to summarise various automated anomaly detection framework and draw logically conclusions at the end and provide a basic approach to design an automated regression detection framework .

*Keywords* — **automated, anomaly detection , regression detection framework .**

--------------------------------------*************************---------------------------------

## I. INTRODUCTION

When a software system starts behaving abnormally during normal operations, it has an immediate impact on customer experience and satisfaction. Performance concerns in a software product are usually serious enough to cause delays in deployment which occasionally is reported as one of the most critical problems of deployed systems.

While the diagnosis of a performance problem can be challenging, it is critical towards the smooth functioning of such systems..Early, informative warnings on significant changes in application performance should help detect anomalies and prevent any major mishap , specially when dealing with sectors like - healthcare, aviation and banking.

To understand the importance of anomaly detection framework, we highlight the importance of anomaly detection framework with the following example- We consider an investment bank, A ,which recently changed its taxation criteria for mortgage bond securities. An erroneous implementation might lead to cases with faulty transactions. Such a software, if it makes through the production stage can cause loss of reputation and business to the bank to avoid these we pass the modified code through a Regression detection framework to find out how these code updates affect the overall trade flow.Given the large volume of trades being executed on a daily basis, the process needs to be automated.

In the next few sections we will be discussing the methodologies of few automated regression detection frameworks and a basic approach to make a simple regression detection framework.

## II. LITERATURE SURVEY

The paper by Hamou-Lhadj[1] talks about a novel framework that allows training and testing of anomaly detection techniques on trace streams in real-time.The TotalADS is divided into 4 parts :

1. The trace management engine deals with the format of the trace, CTF, XML and text based traces are supported.
2. The Anomaly Detection Engine comprises different host- based anomaly detection techniques by using a common interface in Java. TotalADS integrates three different anomaly detection techniques, namely, Kernel State Modeling (KSM), Sequence Matching (SQM) , and Hidden Markov Model (HMM).
3. Trace Inspection Engine comprises of a set of views that can be used to do various different tasks. Few examples include building anomaly detection models, diagnosing traces using models, training and evaluating models on live traces, inspecting individual event details in a trace, understanding control flow of processes, comprehending utilisation of CPUs in a trace at different time, getting event statistics in a trace,

and visualising occurrences of events as histogram of frequencies over a timeline.

4. No sql database is used to store the data.

The initial results from the above method give a lot of false positives as the models don't have enough data but with passing time it gets better and gives the desired results. However the HMM requires other fine tuning to perform as with the case study performed the HMM failed to detect the anomaly.

The paper by Lee[2] tries to eliminate almost all manual overhead in performance testing to detect performance anomalies and find their root causes during regression tests.They have used Statistical Process Control(SPC) and Cumulative Sum (CUSUM) charts.All the changes submitted in the development branch of the code are followed by automated functionality regression testing using the agile development process.Each performance test involves a unique set of metrics where different metrics use different formats for representing the results. The Performance Anomaly Detector for each metric given a test it retrieves the performance data from the QA database, applies SPC charts to it, and displays the results that are considered suspect anomalies.If both the test environment and variation of the measured metrics are stable.

The paper by Memon[3] discusses a GUI control-flow graph (G-CFG) and a GUI call-graph (G-call graph) method approach which contains the following components :

- Test case checker partitions the original test suite into (1) usable test cases, (2) repairable unusable test cases, and (3) unrepairable unusable test cases.
- Test case repairer fixes the unusable test cases by adding and deleting events to the test case in order to match the event sequence with the modified GUI.
  The major drawback of this method is that it cannot be extended for the conventional paper.

The paper by Banerjee[4] developed the following modules :
• Test Case Generator, used to formally define a GUI test

case.

- Test Oracles, used to determine whether or not the software executed correctly during testing.
- Coverage evaluation is a useful guide to additional testing, whether it is done for the next build or for future comprehensive testing.
- Event Coverage tries to exercise individual events in the GUI. The individual events correspond to length one
  event-sequences in the GUI.
- Event-interaction Coverage requires all the sides of the event-flow graph be covered by a minimum of one test suit .

The major drawback of this paper is that it is limited to Smoke Tests.
The papers by Tanja E.J. Vos[5] and by Carol Alexander[6] are overviews of the software regression tests. The first paper has presented three successful instantiations or applications of the framework to validate its applicability and effectiveness. However, the framework needs to be evaluated by many more case studies to validate the completeness of the identified scenarios and the identified variables. On the other hand the second paper has two main objectives first was to identify the most suitable risk models in which to conduct a stress test. They have considered eight relatively well-known and parsimonious risk models, of which four incorporated volatility clustering. The second and major objective of the paper was to develop a methodology for conducting stress tests for a risk model. They proposed a two-stage approach in which an initial shock event is linked to the probability of its occurrence. Finally , the risk

model is used to model the consequences of that shock event.

The paper by G. Rothermel[7] describes a framework for evaluating regression test selection techniques that classifies techniques in terms of inclusiveness, precision, efficiency, and generality. The authors have illustrated the application of this framework by using it to evaluate existing regression test selection techniques.

## III. Regression Detection Framework

The purpose of a regression detection framework is to test whether the modified code affects the other part of software application or not.A regression detection framework is the set of rules, a predefined structure or guidelines that the automated test cases should follow during SDLC (Software Development Life Cycle).The amount of regression testing required depends upon the newly added feature for example , if the feature is large, then there are more chances of the application getting affected. In a regression detection framework we compare the output of two versions of the software. If a change in the software causes an unexpected difference, it would be highlighted. On the event when a difference was expected but not found the same logic is applied. Fig .1 depicts a basic architecture of a Regression Detection Framework.

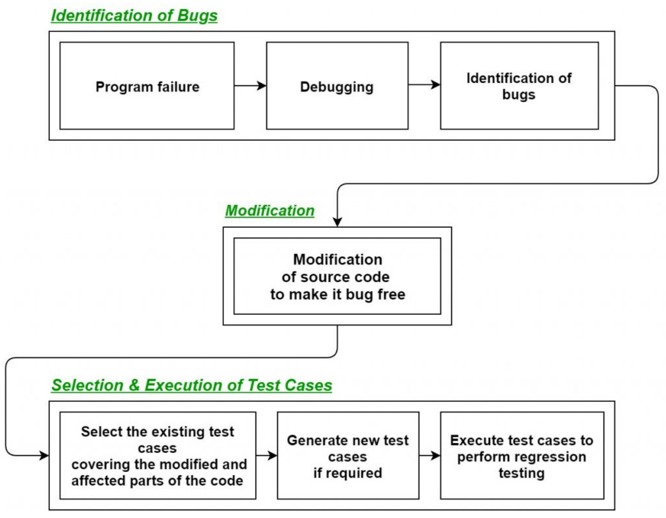
Identification of Bugs
Program failure
Debugging
Identification of bugs
Modification
Modification of source code to make it bug free
Selection & Execution of Test Cases
Select the existing test cases covering the modified and affected parts of the code
Generate new test cases if required
Execute test cases to perform regression testing

Fig 1. Architecture of Regression Detection Framework

In order to do Regression Testing process, we need to find out the type of application that needs to be tested .Then we need to debug the code to identify the bugs. Once the bugs are identified, required changes are made to fix it then we need to do a review of the test cases. Relevant test cases need to be selected from the test suite that covers both modified and affected parts of the code. Regression testing can be made more effective if we select the following test cases :

- Test cases which have frequent defects
- Functionalities which are more useful to the users
- Test cases which verify core features
- Test cases of functionalities which have recently been modified
- All Integration test cases
- All Complex test cases
- Boundary value test cases
- A sample of Successful test cases
- A sample of Failure test cases

Automated testing tools are capable of generating automatically generate a suite of passing unit tests by parsing the source code and determining all possible paths through the code.This reduces reliance on time consuming and expensive system tests also as being far more thorough and identifying the situation of errors more precisely.Automatically running regression tests means frequent test runs which makes bugs easier to repair as regression errors are identified earlier. With the successive runs , test suites become very large and minimising the test suites remains a major challenge.

## IV. Languages used

The following section discuses few languages that can be used to implement a Regression Detection Framework.

### A. Python

An open-source programming language for test automation, machine learning, etc, Python has around 8% of consumers use Python as their language of choice for building test automation tools. Python provides a huge set of libraries and thanks to its simple learning is one the foremost popular language for designing regression detection framework.

### B. Java

Released by Sun Microsystems in the year 1995, Java is a cross-platform object-oriented programming language. Java is a general-purpose, class-based, object-oriented programming language designed for having fewer implementation dependencies.It is built on the principles of object-oriented programming is a general- purpose programming language that's owned by the Oracle Corporation. Around 44% of consumers use Java to create a regression detection framework.

### C. C#

C# is a programming language developed by Microsoft that runs on the .NET Framework. With newer and more powerful languages coming in the market , it is becoming less and less popular but C# still accounts for around 8.8% of the regression detection framework.

### D. JavaScript

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions . It is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative styles.

### E. ShellScript

A shell script is a computer program designed . A Bash script is an example of a shell script, it is a plain text file which contains a series of commands. The flexibility and the ease to work with logs and across various languages and format makes it a popular choice.

## IV. CONCLUSION

After analysing various Regression Detection Frameworks I have come to the following conclusion:

- Existing Regression Detection Frameworks are general and designed mostly for GUIs.
- They can't be integrated with Business Specific Services for which we require the regression test.
- The Existing Regression Detection Framework can't be operated for and end-to-end testing across different services.
- The Existing Regression Detection Framework can't be operated for testing across various formats in which the messages are exchanged across services.

This paper also talks about a basic approach on designing a regression detection framework, the important points to consider while choosing test cases and the challenges faced.

## REFERENCES

1. Abdelwahab Hamou-Lhadj, Wael Khreich and Mario Couture, "Total ADS: Automated Software Anomaly Detection System," 2014 IEEE 14th International Working Conference on Source Code Analysis and Manipulation.

2. Donghun Lee, Sang K. Cha and Arthur H. Lee , "A Performance Anomaly Detection and Analysis Framework for DBMS Development," IEEE Transactions on Knowledge and Data Engineering ( Volume: 24, Issue: 8, Aug. 2012).

3. Atif M. Memon and Mary Lou Soffa , "Regression testing of GUIs," September 2003 , ACM SIGSOFT Software Engineering Notes.

4. A. Memon , I. Banerjee , N. Hashmi and A. Nagarajan , "DART: a framework for regression testing "nightly/daily builds" of GUI applications,"ICSM 2003 authored by A. Memon , I. Banerjee , N. Hashmi and A. Nagarajan.

5. Tanja E.J. Vos, Beatriz Marın, Maria Jose Escalona, Alessandro Marchetto , "A Methodological Framework for Evaluating Software Testing Techniques and Tools," 20.12 12th , International Conference on Quality Software.

6. Carol Alexander , Elizabeth Sheedy ," Developing a stress testing framework based on market risk models," October 2008 , Journal of Banking & Finance.

7. G. Rothermel and M.J Harrold , "Analyzing regression test selection techniques," August 1996 , IEEE Transactions on Software Engineering.

8. Angie Jones , "2020's Most Popular Programming Languages for UI Test Automation," December 16, 2020 [Online] . Available : https:// applitools.com/blog/2020-most-popular-programming-languages-for- ui-test-automation/.

9. Testsigma , "What is Automated regression testing? How and When to implement," [Online] . Available : https://testsigma.com/ regression- testing/automated-regression-testing.

10. Guru99 , "What is Regression Testing? Definition, Test Cases," [Online] . Available : https://www.guru99.com/regression-testing.html.

11. QASystems , "Automated regression testing," [Online] . Available : https://www.qa-systems.com/solutions/automated-regression-testing/

12. S. Yoo , M. Harman , "Regression testing minimization, selection and prioritizations: a survey ," October 2013 , Software Testing, Verification and Reliability.