Load Effectiveness on Coverage-Technique for Test case Prioritization in Regression Testing

Karanam Madhavi, G. Bhavyasri, G. Ramesh, Ch. Mallikarjuna Rao, Lavanya Gottemukkala

***Abstract*: An approach is relied on the potentiality of function-coverage method depends on the correlation value among the coverage and fault rate. As the load increases on the test group, the response of the fault rate and the coverage information have observed. This observation states, while increasing the load the techniques involved are honed the performance of the function coverage technique. Test Case prioritization is an essential aspect for the improvement of functional testing along with coverage data. Test cases grouped and prioritized to lag the regression testing cost and lead to amplify the fault rate. Clustered test scripts are analysed in the increasing of load with the help of specified potential testing tool.**

***Index Terms*: Clustered Technique, Regression Testing, Function Coverage Techniques, Performance Testing, Test Case Prioritization, and Load applied effectiveness.**

I. INTRODUCTION

To estimate the expenditure for regression testing, test case prioritization is the best method to enhance the fault rate. Test scripts arranged in proper way and run in an order. For this purpose, many methods follow the usage of coverage data throughout the test and it is usually said as coverage method [2, 3, 4, 5].

This paper follows the method to arrange the test scripts in order by using cluster test groups and closed structure of dependency [9, 10, 11]. Two coverage data factors TFC and AFC [2, 4, 5] applied to obtain the correlation calculation among coverage and fault rate in test case prioritization.

This Paper is organized as follows, section II shows related work. Function coverage technique used clustered test groups and dependency structure along with load in section III. Section IV shows the results that are evaluated high positive correlation value with increasing load. Section V concludes the proposed work and suggests feasible instructions for further work.

Revised Manuscript Received on May 10, 2019

Dr. Karanam Madhavi, Professor, Department of CSE, GRIET, Hyderabad, Telangana, India.

G. Bhavyasri, Assistant Professor, Department of CSE, Siddhartha Institute of Science & Technology Puttur, Chittoor (Dt)., Andhra Pradesh, India.

Dr. G. Ramesh, Associate Professor, Department of CSE, GRIET, Hyderabad, Telangana, India.

Dr. Ch. Mallikarjuna Rao, Professor, Department of CSE, GRIET, Hyderabad, Telangana, India.

Lavanya Gottemukkala, Assistant Professor, Department of CSE, GRIET, Hyderabad, Telangana, India.

II. RELATED WORK

Many methods has introduced in test case prioritization to make earlier the fault detection [12], [13]. Also Average Percentage Fault Detection (APFD) is to calculate the average fault detection rate [1], [3]. Test cases prioritized by the techniques, also to raise the fault rate detection. These are stratified into three categories i.e., control, statement level, function level [2]. If there were restricted test cases, there would be a low correlation with effectiveness and coverage as moderate one [7], that offered by Laura Inozemtseva and Reid Holmes. An approach proposed by Carlos, states that a hierarchic clustered style and inputs for these are code, fault history and complexity for coverage [9].

However, in the current study, test scripts cluster test group approach and closed-structure of dependency used for segmenting of test scripts and structured based on the coverage data of each test script in addition of load apply. This paper aims to identify what is the efficacy of applied load on coverage techniques in test case prioritization for regression testing.

III. PROPOSED WORK

The process will be start by taking the step of clustering the test scripts in the way of test groups as per functional dependency. These defined as the determination of attributes uniquely of some relationship with each other. Take the biometric example for employee authentication in an organization; employee cannot authorize his thumbprint without a registration of employee thumb input. In this way, there are many test scripts developed to test these kinds.

Figure.1 illustrates the proposed work. Test scripts generated and executed by respective tools and frameworks namely selenium and TestNG. On the other hand, particular load has applied for the current test scripts to check the effectiveness of load on the system. Than the prioritization of test scripts assign by giving inputs as Total Function Coverage (TFC) and Additional Function Coverage (AFC). Test outcomes and reports generated by implementing the priority test scripts. Coverage of test groups will be obtained by the tool namely, EcLemma. Coverage has an essential role in the calculation of correlation.

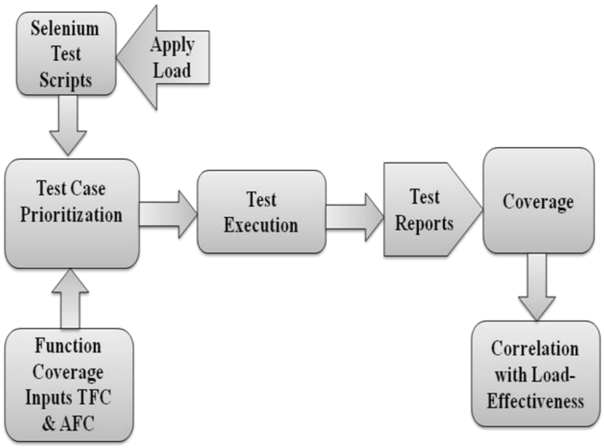

Figure 1: Framework

A. Test Case Prioritization

Test case prioritization offers suitable order to develop the efficiency in reaching of some potential objective and fault detection rate. Coverage-data based Prioritization methods [2, 3, 4, 5] are helpful for above objective and the selective method is Function coverage method [15]. Tester may want to growth the rate of fault detection for test scripts in test groups, the usage of the ones, enlightening in advance the faults in regression testing [12, 13].

Tester may want to enhance fault detection rate in a test groups, using those test groups, enlightening earlier the faults in regression testing. In addition, tester possibly will target to enhance the coverage in at more rapidly under the test in a system, allowing a coverage criterion to meet earlier in the process. These desires stated qualitatively. To determine the achievement of a prioritization method in meeting the sort of aim we should describe the intention quantitatively.

B. Load applied in Test Case Prioritization by Function Coverage Techniques

Clustering test group approach improves the overall performance using function coverage techniques. The coverage data gathered from coverage factors TFC and AFC for all test scripts. By this data, testing condition is “less covered methods are to accept prioritized first” [1, 3]. The approach states that, prioritized test script faces fewer methods. The potentiality of selective method relied on the outcome of the rate of correlation value on the applied load. How will be the behaviour of the test scripts when a particular load applied and how the techniques produce the corresponding outcome for that load against the stability of the technique. Moreover, it is to be either in moderate or high correlation value when the load is increased. Anyhow, this is not forever and that causes to obtain undesirable outcome. As per above statement, if the test scripts obtain the same count of functions that are less covered, the test script with less fault rate is assigned priority first to execute. In this paper, clustering test group mechanism use closed- dependency to arrange the test scripts. Here load applied clustering test group approach exposes a positive correlation among fault rate detection and coverage. The conclusion is the load applied test groups of the current process enhance the overall performance of function coverage techniques. At first, test scripts not prioritized and tested without priority. The flanking test scripts as per the condition, arrange the first test group and the remaining closets arranged with new test groups. Following the segregation of test groups, test scripts arranged by using the closed structure of dependency. Oder assignment on each test group as follows.

S1: Test scripts placed on each test group through the dependency type as closed one.
S2: Apply prioritization technique on testing through the factors of function-coverage:

(TFC) Total Function Coverage [1, 2]
(AFC) Additional Function Coverage [1, 2]

C. Cluster Test Groups with applied load input.

Input: Test Suite $T_i = \{TS_1, TS_2, \dots ,TS_n\}$
Algorithm#1:
Start
Set T_k empty
For Every test Script
Cluster Test Group $TG_i = TS_i$ by TF_i
Input AT_n in TG_i as applied load
End for
Arrange TS_i in TG_i, in appropriate way as per TF_i.
$T1 = TG_i$
Assume T_k=T1
Stop
Output: Clustered Test Groups having Test Scripts T_k

Algorithm#1 specifies the clustering test groups in appropriate order accord ing to function dependency TFi. Applied Load AT_n given as input to remaining test scripts and arranged in test group by the condition. Algorithm#2 says that, source data TFC, AFC from function coverage technique along with applied load AT_n collected into Test Groups $TG'_{i,}$ using that data, test scripts TS_i & TS_i (AT_n) prioritized in $T_{j.}$

D. Test Scripts prioritized in Test Groups Using Function Coverage Techniques on Applied Load

Input: T_{TFC}, T_{AFC} factors from T_k, which is a clustered test suite.
Algorithm#2:
Start
Set T_j empty
For Every $TG'_i = AT_n$
Arrange TS_i & TS_i (AT_n) in TG'_i, as increasing order based on coverage factors TFC, AFC with additional load.
End for
$T1 = TG'_i$
Assume T_j = T1
Stop
Output: Load applied test groups with priority of test

scripts in T_j.

IV. RESULTS

Test scripts in clustered test groups rely on closed-dependency shown in table1. Test scripts automated by the selenium tool and test scripts executed by TestNG framework. Test scripts in clustered test groups prioritized mostly by the coverage information i.e., TFC and AFC and computed as follows:

TFC = Number of functions executed totally.
AFC = Number of additional functions covered.
Test Group Defects = Defect count in cluster test group.

Table1: Clustered Test Scripts

Test Groups (TG_i)	**TG_1**	**TG_2**	**TG_3**
Test Scripts (TS_i)	TS1-TS1 0	TS1-TS20	TS1-TS30

Table2: Coverage Factor values for test scripts with Clustering Test Groups.

Test Scripts (TS_i)	**TS_{TFC}**	**TS_{AFC}**	**Clustering Test Groups (TGi) Defects**
Test Script3	12	1(f2)	
Test Script4	16	1(f12)	
Test Script5	17	1(f14)	Test Group1
Test Script7	8	1(f5)	
Test Script10	7	1(f7)	
Test $Group_2$	7	1(f6)	Test Group2
Test Script21	4	1(f1)	
Test Script22	6	1(f2)	Test Group3
Test Script26	10	1(f10)	

Table3: Group-Wise Test Scripts with Priority

Test Groups	**Prioritized Test Scripts**
Test Group1	TS8, TS10, TS7, TS1, TS3, TS2, TS9, TS4, TS5, TS6
Test Group2	TS8, TS10, TS7, TS1, TS3, TS2, TS9, TS4, TS5, TS6, TS19, TS20, TS18, TS13, TS15, TS16, TS14, TS17, TS12, TS11
Test Group3	TS8, TS10, TS7, TS1, TS3, TS2, TS9, TS4, TS5, TS6, TS19, TS20, TS18, TS13, TS15, TS16, TS14, TS17, TS12, TS11, TS21, TS24, TS22, TS28, TS30, TS27, TS26, TS25, TS23, TS29

The test scripts prioritized based on the function-coverage technique [15]. In the Table 2, coverage factors computed for entire load of scripts that in appropriate order. By the outcome, priority condition is "less covered methods are prioritized first". With this main idea, test scripts prioritized and tested. As already stated, the situation with duplicate function count value of scripts covered, low fault rate scripts assigned as priority one. In addition, additional function coverage achieves the high pass rate than the other methods [16]. Test scripts that provide function coverage data as follows the table.

Table 4: Fault Rate for Test Groups

Clustered Test Groups	**Failed Test Scripts**	**Fault Rate**
Test Group1	TS3, TS4, TS5, TS7, TS10	0.014
Test Group2	TS3, TS4, TS5, TS7, TS10, TS17	0.015
Test Group3	TS3, TS4, TS5, TS7, TS10, TS17, TS21, TS22, TS26	0.021

Average Percentage Fault Detection (APFD):

APFD is to measure the detection of fault rate in appropriate order [1, 14]. Prior fault detection indicated by the high assessment of this metric.

$$APFD = 1 - \frac{2+1+1+1+4}{(22)(9)} + \frac{1}{44}$$

$APFD_{TG1} = 0.95$
$APFD_{TG2} = 0.97$
$APFD_{TG3} = 0.96$

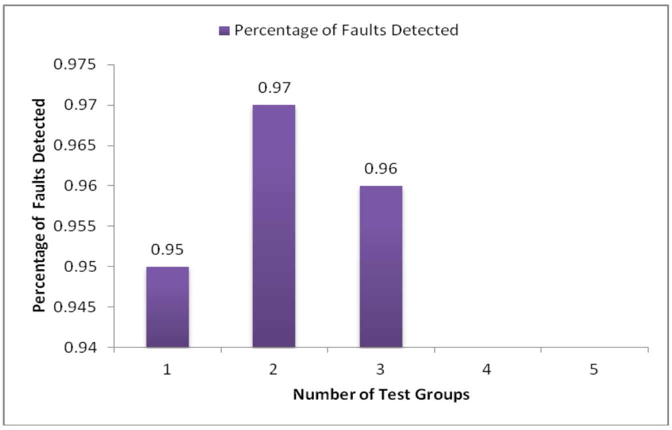

Figure 2: APFD for Load applied Test Groups

A. Correlation

For every test script in a test group, coverage data computed through the EcLemma. Presented one is the prospect to prove the correlation in the positive linear value [7, 8, 16].

Table5: Coverage rate for a variety of categories

Coverage category	Coverage (%)		
	Test Group1	Test Group2	Test Group3
Statement	73.4	83.5	89.1
Branch	50.0	50.0	50.0
Method	90.5	93.5	95.3
Complexity	86.4	90.6	89.1

Once the computation of coverage data and fault rate has done, using pearson correlation the value of correlation will be computed. The corresponding inputs are coverage as 'Cov' and fault rate 'fr'.

Table6: Fault Rate in Test Groups

TG_1	TG_2	TG_3
0.0146	0.0153	0.021

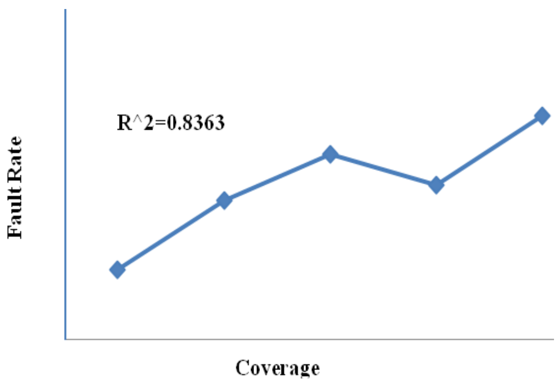

Figure.3: Correlation of Existing work

Figure.4 says that the resulted correlation is reflection of positive value of correlation 97% that states load effectiveness does not lag the technique applied in this work [17, 18].

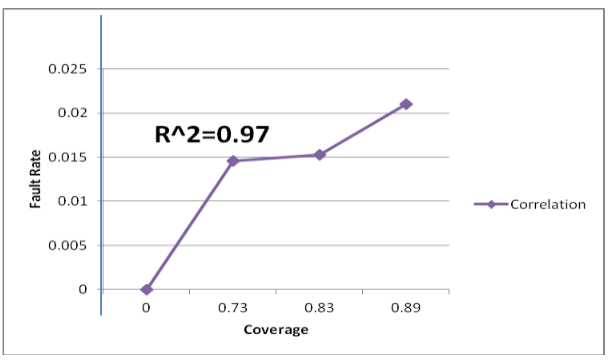

Figure.4: Load applied correlation value for proposed one

The Figure. 4 illustrates the varieties of coverage and fault rate for test group scripts with function level techniques by increasing load as more test scripts. However, in case of existing work, in Figure..3 the rate of correlation shows less when compare with proposed one that is increased rate of correlation. Enhanced correlation rate says that more potent coverage sorts provide a few greatest data in test groups with a growing load.

V. CONCLUSION & FUTURE WORK

The proposed work projected such as the clustered and closed-dependency structure for test case prioritization with function level technique. In addition, the load has applied to ensure the outcome rate of correlation. The results provide the high rate of positive correlation value between coverage and fault rate and proved experimentally with regard to load increase, the performance in coverage-based technique is still efficient, because in existing work, the coverage and fault are correlates in moderate or positively as per size. Here the process taken to check the correlation will be either positive rate or negative rate when load applied for the current system. Then the outcome shows the rate of correlation with increased positive value than the existing rate with comparison of both.

Improved rate of fault detection in an average has computed through APFD. Coordinates on correlation show a positive rate. The feasible enhancement shows the capacity of programmed test scripts to fulfil overall potential objective in reliability and further coverage strategies in destiny. Ontology method is a simple and best suitable for prioritization in regression testing with huge information system maintenance can be considered for future work.

REFERENCE

1. G. Bhavyasri and A. Ananda Rao, "Enhancing the Performance of Coverage-Based Techniques in Test Case Prioritization", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol 5, 2017, pp. 468-473,.
2. Elbaum, S., Malishevsky, A. G., &Rothermel, G. "Prioritizing test cases for regression testing". ACM , Vol. 25, No. 5, 2000, pp. 102-112.
3. Eghbali, S., &Tahvildari, L. "Test Case Prioritization Using Lexicographical Ordering". IEEE Transactions on Software Engineering, 42(12), 2016, 1178-1195.
4. Elbaum, S., Malishevsky, A. G., &Rothermel, G. "Test case prioritization: A family of empirical studies". IEEE transactions on software engineering, 28(2), 2002, 159-182.
5. Rothermel, G., Untch, R. H., Chu, C., &Harrold, M. J. "Prioritizing test cases for regression testing. IEEE Transactions on software engineering", 27(10), 2001, 929-948.
6. Akbar SiamiNamin, James H. Andrews, "The Influence of Size and Coverage on Test Suite Effectiveness" Department of Computer Science", Texas Tech University at Abilene, University of Western Ontario London, Ontario, Canada.
7. Inozemtseva, L., & Holmes, R.. "Coverage is not strongly correlated with test suite effectiveness."In Proceedings of the 36th International Conference on Software Engineering, may 2014, pp. 435-445.
8. Srikanth, H., Williams, L., & Osborne, J. "System test case prioritization of new and regression test cases".In Empirical Software Engineering, International Symposium, Nov 2005, pp. 1-10.
9. Carlson, R., Do, H., & Denton, A. "A clustering approach to improving test case prioritization: An industrial case study". (ICSM), 2011 27th IEEE International Conference on Software Maintenance , Sep 2011, pp. 382-391.
10. Arafeen, M. J., & Do, H. "Test case prioritization using requirements-based clustering". (ICST), 2013 IEEE Sixth International Conference on In Software Testing, Verification and Validation, March 2013, pp. 312-321.

11. Miller, T. "Using dependency structures for prioritization of functional test suites". IEEE Transactions on Software Engineering, 39(2), 2013, 258-275.
12. Jung-Min Kim and Adam Porter, "A History-Based Test Prioritization Technique for Regression Testing in Resource Constrained Environments", 1CSE'02, 2002, pp.119-129.
13. Zheng Li, Mark Harman, and Robert M. Hierons, "Search Algorithms for Regression Test Case Prioritization", IEEE Transactions on Software Engineering, Vol. 33, NO. 4, 2007, pp. 225-237.
14. Shin Hong, Matt Staats, Jaemin Ahn, Moonzoo Kim and Gregg Rothermel, "The Impact of Concurrent Coverage Metrics on Testing Effectiveness", 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation, 2013, pp. 231-241.
15. Paul, Ohba and Joe, "Coverage Measurement Experience during Function Test", IEEE, 1993, pp. 287-301.
16. Sangmin Park, Ishtiaque Hussain and Kunal Taneja, "CarFast: Achieving Higher Statement Coverage Faster", SIGSOFT'12/FSE-20, 2012, pp. 1-11.
17. Hyunsook Do and Ladan Tahvildari, "The Effects of Time Constraints on Test Case Prioritization: A Series of Controlled Experiments", IEEE Transactions on Software Engineering, Vol. 36, NO. 5, September/October 2010, pp. 593-617.
18. Nadia Alshahwan and Mark Harman, "Coverage and Fault Detection of the Output-Uniqueness Test Selection Criteria", ISSTA '14, July 21-25, 2014, pp. 1-12.

AUTHORS PROFILE

Dr.K.Madhavi**,** working as a Professor in Computer Science and Engineering Department, Gokaraju Rangaraju Institute of Engineering and Technology. She has completed her B.E in 1997 from Kuvempu University, M.Tech from JNTUA in 2003 and awarded Ph.D from JNTUA in 2013. She has 20 years of teaching experience. She has published several papers in reputed international journals and international conferences. Her research interest includes software engineering, Model Driven Engineering, Data Mining, Mobile Software Engineering, and Educational Technology. She is life member of CSI and ISTE.

Dr. G. Ramesh is currently working as an Associate Professor in Department of Computer Science & Engineering, GRIET, Hyderabad. He received his B.Tech degree in information technology from RGMCET, Nandyal, Kurnool Dist. Andhra Pradesh, and He received his M. Tech degree in software engineering from JNTUA College of Engineering, Ananthapuramu, Andhra Pradesh, India, He received his PhD from JNTUA, Anatapuramu, Andhra Pradesh, India. His main research interest includes software engineering and big data. He is in teaching since 2008. He has published a several papers in various international journals/ conferences. He is life member in IAENG.

Dr. Ch. Mallikarjuna Rao received the B. Tech degree in Computer Science and Engineering from Dr. Baba Sahib Ambedkar Marathwada University, Aurangabad, Maharastra in 1998, M.Tech Degree in Computer Science and Engineering from JNTU Ananthapuramu, Andhrapradesh in 2007 and Ph.D in Computer Science and Engineering from JNTU , Ananthapuramu, Andhrapradesh in 2016 Currently he is working in " Gokaraju Rangaraju Institute of Engineering and Technology", Hyderabad. Telangana , India. His area of Interests are Data Mining, Bigdata and Software Engineering.

G.Lavanya working in Computer Scince and Engineering Department, Gokaraju Rangaraju Instittute of Engineering Technology, Hyderabad. She has completed her B.Tech in 2010, M.Tech from JNTUH in 2012.She has 4 years of teaching experience. Her research interest include sofware engineering, Model Driven Engineering.