# Configuration Navigation Analysis Model for Regression Test Case Prioritization

**Manas Kumar Yogi**[1]
[1]Department of Computer Science and Engineering
[1]Ellenki Engineering College, India

*Abstract*—Regression testing has been receiving increasing attention nowadays. Numerous regression testing strategies have been proposed. Most of them take into account various metrics like cost as well as the ability to find faults quickly thereby saving overall testing time. In this paper, a new model called the Configuration Navigation Analysis Model is proposed which tries to consider all stakeholders and various testing aspects while prioritizing regression test cases.

*Key words:* Regression Testing, Analysis, Stakeholder, Rank, Navigation, Test case Prioritization.

## I. INTRODUCTION

Any type of software system tends to evolve as it is adapting to the dynamic environment, modifiable needs, new concepts and new technologies. So, it will grow in the number of components, functions, and interfaces .Existing functionality may be expanded for uses beyond their original design. Thus, changes to the software are unavoidable. According to industry estimates, it has been that software maintenance consumes nearly 50 to 80% of the total software cost. A major stage of maintenance is the retesting of the software. The reduction in cost of maintenance is possible by adopting efficient regression testing strategies.

Regression testing is done after the bug fixing work has been completed. Some test cases from the original suite of test cases are selected to validate the system functionality. Thus Regression testing helps in keeping the confidence of the testing team high and also helps in tracking the bugs effectively. A number of testing tools exist to help in software development; but, few of them can be applied directly to regression testing. Among those which can be useful for regression testing, most have no more than the capability to store the previous tests and rerun them after every modification. They are not capable of any intelligent test selection and they cannot estimate the required testing effort.

The Regression testing strategies at unit level attempt to select a subset of the previous test for execution and do not repeat all the previous tests. The test cases store some data so that after specific changes are made to the system, those test cases can be easily identified. But the most important aspect is consideration of saving of cost while retesting some old test cases. This paper proposes a novel approach named as the Configuration Navigation Model. In section II, the overview of existing regression test prioritization strategies is presented. Section III discusses the proposed model in details and section IV indicates the Challenges posed for the model and concludes with the plan for future research in section V.

## II. OVERVIEW OF EXISTING REGRESSION TEST PRIORITIZATION STRATEGIES

Many models are used for prioritizing regression test cases. They may be based on code coverage of a program segment. Metrics like weighted average of the percentage of branch covered (APBC), percentage of decision covered (APDC) and percentage of statement covered (APSC) help in ranking the test cases. But the severity of faults is generally not an effective factor in this type of strategy. The main drawback is that it does not show encouraging results when time constraints are bound to it.

Another approach is using the Models to generate regression test cases like use case diagrams, class diagrams, state machine diagrams, activity diagrams. As, it is known that in UML artifacts are interrelated with each other so any change in one diagram forces changes in the other. For instance, if in a class diagram we change a specific operation then the corresponding message in the sequence diagram also changes. The behavioural models also depend on each other and due to system modification it becomes really difficult to carry out an efficient test analysis. This shortcoming creates a type of hindrance to independability of test case design.

The Risk based regression testing takes into account the severity of the fault by assigning an appropriate cost to the fault and according to probability of occurrence of the fault a risk exposure factor is calculated as:

$$RE(f) = P(f) * C(f) \qquad (2.1)$$

Another approach for regression testing of service centric systems is the Quota Constrained where the concept of Request Quota is used along with the technique of Inter Liner Programming (ILP) for test case selection. It uses a multi stage iterative process of Time Slot Partition, Test case selection and Prioritization for each time Slot and Information Refreshing. But this approach is developed only for Web Services. The drawback of this model is that any constraints other than request quota are not handled.

The History based Test prioritization strategy takes into account the historical data of test execution. Here a historical value based approach with the historical data to estimate current cost as well as the fault severity. The cost cognizant, test case prioritization uses the function level gratuity and the historical information of the cost of the test cases .The fault severities values help in producing an effective APFD (Average Percentage of Faults Detected) metric. The drawback of this approach is that only the effect of last execution of test cases is used to calculate the probability of selecting test cases and that too in a binary fashion. Another drawback is that this approach behaved differently in a constrained environment and for unconstrained environment. For lengthy regression testing process the historical based concept was efficient but it was not cost effective for small software systems.

The Genetic Algorithm based test prioritization criteria uses genetic operators like crossover, mutation, selection, transformation operators to reorder a regression test case suit. This approach is effective in case of large number of test cases interdependent on each other. Research is going on in this regard. But according to experimental studies, the Genetic based approach was found to produce better results than other evolutionary search based techniques which can be applied for test prioritization.

## III. CONNECTION NAVIGATION ANALYSIS MODEL

### A. *Overview*

The proposed technique in this paper is composite in nature. It considers equal participated of all who are in need of the services provided by a software system so that the failure rate of the system is decreased to as much as possible. The Novel approach proposed in this paper has 3 stages of analysis each depending on other due to requirement traceability and other essential factors that cannot be ignored while regression testing. The model is named as Configuration Navigation as the software system is tested in the way it is built considering a conventional linear sequential model for development. In each navigation stage we consider various factors which are helpful in enhancing the effectiveness of regression testing process.

### B. *Stakeholder Analysis Stage*

The various user categories according to their specific business needs will rank the test cases after receiving a stable build. The build is delivered to stakeholder after bug fixing is completed. Each test case is ranked as follows,

| TC ID | Rank (1-10) |
|---|---|
| 1 | … |
| 2 | … |
| 3 | … |

Table. 1: Stakeholder's Rank Table

The Rank 1 indicates highest test case priority and 10 indicate the lowest priority.

### C. *Business Analyst Evaluation*

The business analyst decides strengthening the ranks of the test cases obtained after stakeholder analysis. Each user category is pre-assigned a Weightage depending on pragmatic issues. The test case rank evaluation is done as shown below.

| TC ID | Rank (1-10) | Stakeholder Weightage (1-5) | Cumulative Value |
|---|---|---|---|
| 1 | | | |
| 2 | | | |

Table. 2: Business Analyst Evaluation Table

The Cumulative value is computed with the following equation:

$$Cumulative\ value = Rank\ * Stakeholder\ Weightage \tag{2.2}$$

If two test cases have same Cumulative values then the Analyst performs Cost/Benefit evaluation and then decides the prioritized order. Stakeholder Weightage values are given on a scale of 1-5 with 1 indicating the highest Weightage and 5 for lowest Weightage. The lower cumulative values are preferred over the higher. Then these values are passed on to the regression testing team.

### D. *Regression Testing Team Analysis*

The Cumulative value obtained from the earlier stage is used to prioritize test cases in an order such that overall testing effort and cost is reduced. It is done by considering various constraints like time taken to design a test case, complexity in test case design, amount of testing resources consumed by a test case and the time taken to fix the bug. .The team then arrives at a value which gives a clear understanding regarding the order in which the test cases may be executed effectively. The technical complexity in test case design is represented by CXT, The time to design a test case is denoted by DT, the amount of resources consumed is symbolized by RCT and finally the time to fix a bug is denoted by FT. The final Connection Navigation value (CN) is calculated as follows:

$$CN = \frac{Cumulative\ Value \times (CXT + DT + RCT + FT)}{Total\ Number\ of\ Test\ Cases} \tag{2.3}$$

The test cases are then arranged by increasing order of CN values which gives a prioritized order of regression test cases.

## IV. CHALLENGES IN IMPLEMENTATION

The approach presented in this paper has to be experimententally validated and for that some challenges need to be faced in each of the 3 stages mentioned above.

*Stage 1:* There may be ambiguity in assigning ranks by individual user within a particular user category. This is due to the fact that requirement perception of each individual may not be same. One may assign a lower rank for the same test case which might have been assigned a higher rank by another. This dissimilarity may be reduced by application of additional constraints but still it poses a challenge.

*Stage 2:* The Business analyst faces challenge while performing Cost/Benefit analysis which may or may not equate with those present in the predefined baselines. In some cases multiple test cases may have same cumulative values but still a priority must be assigned to the test cases which are distinct. This shortcoming can be reduced only by careful analysis of the baseline document against the test cases.

*Stage 3:* The greatest challenge lies in this stage. The technical complexity in designing, a test case directly proportionate the time for test case design. If the system under development is entirely new domain for the team then these Weightage values will be naturally high. Resources consumed for the test cases depend on the testing model adopted by the testing team. The time to fix the bugs depends on whether manual testing is done or automation tools are applied for the system.

## V. CONCLUSION AND FUTURE RESEARCH

The relationship between the various cumulative values and CN values can be reaffirmed only by experimental established for large scalable software systems. Highly reliable software is always desirable but cost must be reasonable. Hence this paper tries to establish a relationship between all those who influence the construction of a software system with regression testing strategies. The proposed model can be used to downscale the overall testing budget by effective analysis.

## REFERENCES

[1] S. Elbaum, A. Malishevsky, and G. Rothermel, “Incorporating varying test costs and fault severities into test case prioritization”, Proceedings of the International Conference on Software Engineering, May 2000.

[2] G. Rothermel, R. H. Untch, C. Chu, and M. J. Harrold, “Prioritizing test cases for regression testing”, IEEE Transactions on Software Engineering, Vol.27, No.10, pp. 929-948, Oct.2001.

[3] H. Do, G. Rothermel. “A controlled experiment assessing test case prioritization techniques via mutation faults”, Proceedings of the International Conference on Software Maintenance (ICSM),pp.411-420, 2005.

[4] S. Yoo, M. Harman, "Regression Testing Minimization, Selection and Prioritization: A Survey", Software Testing, Verification and Reliability, Wiley Interscience, 2010.

[5] W. E. Wong, J. R. Horgan, S. London, and A. Aggarwal, “A study of effective regression testing in practice”, Proceedings of the Eighth International Symposium Software Reliability Engineering, pp. 230-238, Nov. 1997.

[6] R. C. Bryce, A. M. Menon, “Test Suite Prioritization by Interaction coverage”, Proceedings of the workshop on domain specific approaches to software test automation (DOSTA), ACM, pp. 1-7, 2007.

[7] F. Belli, M. Eminov, N. Gokco. “Coverage-Oriented, Prioritized Testing-A Fuzzy Clustering Approach and Case Study”. In: Bondavalli. A., Brasileiro, F. Rajsbaum, S. (eds.) LADC 2007, LNCS, Springer, Heidelberg, Vol. 4746, pp. 95-110, 2007.

[8] H. Do, S. Mirarab, L. Tahvildari, G. Rothermel, "An Empirical Study of the effect of time constraints on the cost benefits of regression testing" Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software Engineering, pp71-82, 2008.

[9] B. Beizer, Software Testing Techniques, New York: Van Nostrand Reinhold, 1990.

[10] D. Binkley, ªSemantics Guided Regression Test Cost Reduction, IEEE Trans. Software Eng., vol. 23, no. 8, pp. 498-516, Aug. 1997.

[11] T.Y. Chen and M.F. Lau, “Dividing Strategies for the Optimization of a Test Suite,” Information Processing Letters, vol. 60, no. 3, pp. 135-141, Mar. 1996.

[12] H. K. N. Leung and L. White, “Insights Into Regression Testing”, Proc. Conf. Software Maintenance, pp. 60-69, Oct. 1989.

[13] W. E. Wong, J. R. Horgan, S. London, and A.P. Mathur, ªEffect of Test Set Minimization on Fault Detection Effectiveness,º Software Practice and Experience, vol. 28, no. 4, pp. 347-369, Apr. 1998.

[14] J. Karlsson, K. Rayan , “A Cost value approach for Prioritizing requirements,” IEEE Software Vol 14, NO 5, 1997.