Cost Optimization Techniques for Software Testing

Avinash Yadav ,GITS Gwalior

***Abstract*-- Software Testing is an area of software engineering where software is to be tested with several techniques. As it is well known that software testing is supported by various test cases. All test cases are used to manipulate software for proper utilization of resources and minimize the software cost as well as less execution time. In this paper, it has been focused on the various existing software testing and optimization techniques, proposed by various authors. To optimize the software, Identification, characterization and automatic prioritization of test cases in software testing are being used by optimization techniques**

***.Keywords*-- Software testing, Optimization techniques, Test case selection.**

I. INTRODUCTION

Software testing is the most critical part of the Software Development Life Cycle. Efficient software testing will contribute to the supply of consistent and good quality oriented software, more satisfied users, lower maintenance cost, and correct and reliable result.

However, unproductive testing will lead to the differing results; low quality products, unsatisfied users, high maintenance costs, unreliable and inaccurate results. Thus, software testing is an essential and important activity of software development cycle.

"Software Testing is the procedure of executing a project with the aim of finding mistakes"[1]. The significance of testing can be understood by the way that "around 35% of the elapsed by time and more than half of the aggregate expense are using in testing programs" [2-3]. Software is relied upon to work, meeting customer's changing requirements time to time and predictably.

Prior software frameworks were utilized for back office and non basic operations of associations. At the present more critical applications are executed globally.

This increased expectation for error-free execution of software has improved the interest for quality result from software merchant [4].

During the testing Bach's test approach model [7] is responsible for quality factors with the guiding principle of testing maturity model [6]. Where Key Process Areas (KPAs) is the main quality and consistency parameter, which are to be achieved by software testing process.

Recently, metaheuristic search techniques are used for the automatic generation of test data. Previously automated test generation procedure has been restricted and test data generation is very complex process [8].

The possible cost optimization from managing software bugs within a development, rather than the consequent cycles, has been evaluated at about 40 billion dollars through the National Institute of Standards and Technology [9].

II. SOFTWARE TESTING

Software Testing is an activity that produce a quality report depends on given input. Validation is executed not in favor of requirements and product specifications

2.1 Testing Techniques

Testing methods are based on two techniques: Path based Testing and Functional based Testing. The main aim of Path testing is to "implement" program graph paths. And the aim of Functional testing is to prove functional requirements which are depending on software. Such methods are enforced by several studies helpful a coupled association between software involvement and bugs individuality. Where the program formation is totally depends on the possible paths. But complete testing is impossible, and then selects a best test case.

2.2 Testing strategies

Here are "static" testing strategies, which do not involve program execution. They largely lead themselves to a depth code assessment and reach, however, preliminary and restricted. Also there are several "dynamic" strategies from which the most significant appearance to be clearly.

2.2.1 Path testing

In the path testing all paths should be executed *at least* once. Here the number of paths possible and this strategy can be applied only with extreme reduction.

2.2.2 Branch testing

It requires every branch in a system to be checked *at least* once; this is the easy and best known methodologies and it is less efficient than path testing because bugs depends extra branches combination than a single one.

2.2.3 Functional testing

Aim of functional testing is to check software functionalities independently from its inner structure. In it test case selection to be made upon functional connections between input and output values.

© 2016 IJAIR. All Rights Reserved

Being obviously all the possible I/O interaction near to endlessness, here too there is the issue of removing a suitable test case set.

2.2.4 Structured testing

It estimated path testing, in this the problem under test to be separated into an order of useful modules, in which the single modules are tested first and after that their integration within the whole system is checked. This testing increases branch testing consistency because it permits confirming branch combination. Now a day's no one testing procedure is complete and reliable, still, no one of them insurances software accuracy.

2.3 Approaches for testing optimization models

The behavior of tests usefulness with respect to assigned resources is asymptotic. Means that, among every possible test, only some portion of them is economically helpful.

Resource allocation is the testing optimization issue. Let us assign to every program's runable path its test implementation cost and its test implementation value relevant to possible found bugs. The tests implementation total value is the sum of selected paths tests implementation, and the tests implementation cost is the sum of their corresponding costs. There are two probable approaches of testing optimization are follows:

•*Limited resources model.* To exploit the tests execution value, given a highest execution cost.

•*Given quality model.* To reduce the tests implementation cost, given a lowest execution value.

Software testing techniques:

Selecting test inputs, running the inputs on the product for testing, and assessing the accuracy of the desired outputs. Programming testing happens ceaselessly amid the product improvement life cycle to identify blunders as right on time as could reasonably be expected and to guarantee that current programming don't break the product. Determination and prioritization of experiments are the two noteworthy answers for the issue of experiment streamlining. Any experiment prioritization calculation can be utilized an experiment choice Algorithm. Test suite minimization is a determination of littlest subset the experiments.

i) Bee Colony Optimization (BCO):

Honey bee state is advancement procedure which is based upon the common marvels and discovers ideal arrangement toward the end. It is self arranging system. Two sorts of honey bee accessible in the apiary. Upon these two honey bees all the sustenance gathering procedure is depended.

Scot honey bee is going outside in the inquiry of the sustenance and return to the apiary when out of vitality. Waggle move is performed fit as a fiddle of digit 8 by the scout honey bee in the apiary to impart with the forager honey bee. With the correspondence forager honey bee came to think about the best nature of nourishment toward sun then take after the same way for the accumulation of the sustenance. Scout honey bees investigate the way where as forager honey bees misuse the way. A BCO calculation is utilized to discover greatest number of issues.

ii) Ant Colony Optimization (ACO):

It is a meta-heuristic strategy which is based upon the normal wonders. ACO is a probabilistic method which gives arrangement by utilizing past results. In this procedure every subterranean insect take after various way to reach to the destination and discharge pheromone fluid while in transit to destination. The way which has the most noteworthy fluid pheromone is considered as the briefest way and every single other insect take after the same way. So pheromone fluid is utilized to pull in the other insect and overhaul the most recent data about the way.

iii) Genetic Algorithm (GA):

Hereditary calculation, a versatile hunt method is presented by John Holland comprehensively examined by Goldberg and De Jong. It is an improvement strategy which gives close ideal answer for NP-difficult issues. GA is connected to take care of numerous issues like voyaging salesperson issue rucksack issue and so on. Hereditary Algorithm depends on the thought on the normal development. The establishment of GA lies on the idea of the survival of fittest into an answer space. Every cycle of GA procedure incorporates introduction (encoding), determination taking into account wellness capacity, multiplication utilizing hybrid or change. The cycle is rehashed till an answer is found that fulfills the base criteria or an altered number of eras has been come to.

iv) Particle swarm optimization (PSO):

The methods portrayed are: Particle Swarm Optimization (PSO). The PSO is a worldwide enhancement calculation taking into account heuristic hunt. The thought was given by John Kennedy and Eberhert, in 1995 , subsequent to watching the gathering of creatures, group of feathered creatures and fishes, where every individual takes after the way of ―global best molecule inside of its populace, e.g. in sea, while hunting down great sustenance source, a school of fish. Within story: each fish is watching its neighbors' position and speed then contrast it and worldwide best position and speed.

© 2016 IJAIR. All Rights Reserved

The best position and speed is picked and upgrades are made by people in their position and speed. Thus, every fish unites towards the best position after alteration in its speed, which moves towards the nourishment speedier.

III. Problems With Existing Methods

Experiment Selection is required to choose specific Test suites or test case(s) keeping in mind the end goal to accomplish blame free framework with insignificant cost and time utilization. So we require a viable component of selecting test suits or test case(s).

There is a space among generation of test group, reprocess test suite, and test case prioritize. Organizations usually maintain the test suites for reuse, on the grounds that it represents half of the maintenance cost.

Subsequently, there is a need to expand the viability for various test cases, which will give the test group with sound experiment prioritization at the very least cost.

For finding suitable test cases there is no complete method till date. For search-based testing, there is still space including flag and enumeration variables are disorganized control flow; and state behavior.

Besides, there may be a multiplicity of other logics as to why test records cannot be found with easily for project architecture utilizing search-based procedure.

There are so many gaps in research in the field of search based black box testing analyze to white box testing.

Operative or Functional testing can be derived from different forms of requirement. For tests resultant in this way, a present boundary to finish is the way that a mapping should be given from the abstract model of the specification to the actual form of the execution.

And for system tests, a possible issue is the size of the search space. Effort on Software testing procedure is approximately 40% (SEI) of the aggregate development cost [8].

Reason of this attempt in terms of money, recruitment level, and resources needed, etc. is not understood to the client. It is planned to give accurate justification for the efforts and cost involved.

For determining these parameters Statistical methods with similarly statistical software will be helpful. A trustable and correct estimate of software development requirement had always been a task for both the software companies and academic world. Now we propose to improve software testing effort requirement estimation with appropriate optimization methods. For improving effort estimation Metaheuristic technique also helpful. Significance of software test procedure in software quality and consistency requires accurate quantification and measurement.

So propose to plan a structure for quality and consistency factors to verify and ensure error free, efficient and trustable software. Industries need the testing management for business success; it is more employed to enable societies to achieve sustainable competitive benefits. On the other hand, no one approaches has been developed until now which permit societies to find their present state of testing management on a process and project level, which can obtain basic steps for advance development. Now plan to develop mathematical models. Presently no common technique to measure the difficulty of software testing. Efforts will we made to observe a concluding size method to measure the difficulty of software testing.

IV. Conclusion

The principle objective of exploration is to study on different methods of Software Test Case Selection and Prioritization in Ant Colony Optimization and in addition brief correlations of ACO calculation with GA, SA and PSO calculations and testing methodologies.

REFERENCES

[1] The Art of Software Testing,John Wiley and Sons, Myers G. J., NY, USA, 1979.

[2] Software Testing Techniques, Beizer, B., Van Nostrand Reinhold, USA 1990.

[3] ”Testing: A Roadmap”,M**.** Harrold, International Conference on Software Engineering, (ICSE 00), Limerick, Ireland, 2000.

[4] ”Software Testing: Principles and Practices”, Srinivasan D., Gopalaswamy R., Pearson Education, New Delhi India, 2006.

[5] “The Growth of Software Testing”,Gelperin D., Hetzel B., Communications of the ACM, 1988, 31: 687-695.

[6] Guidelines of Testing Maturity, Professional Tester, Vol. 3, Issue 1, March 2002.

[7] “Software Testing as a Social Science”,CemKarner, IFIP Working Group10.4, Florida Institute of Technology, Siena Italy, July 2004.

[8] ”Search-based Software Test Data Generation: A Survey, Proceedings in Software Testing”, Phil McMin, Verification and Reliability, 2004, 14:105-156.

[9] “High Volume Software Testing Using Genetic Algorithms”, D. J. Berndt and A. Watkins, Proceedings of the 38th (IEEE) Hawaii International Conference on System Sciences, Waikoloa, Hawaii, Jan 2005.

[10] NachiappanNagappan; Laurie Williams; Jason Osborne; MladenVouk; Pekka, Abrahamsson (2005): Providing Test Quality Feedback Using Static Source Code and Automatic Test Suite Metrics. Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering, Chicago, pp85 - 94.

[11] Mark Last; Menahen Friedman; Abraham Kandel (2003): The Data Mining Approach to Software Testing. Proceedings of KDD, ACM Press, Washington.

[12] D. Berndt; J. Fisher; L. Johnson; J. Pinglikar; A.(2003): Watkins: Breeding Software Test Cases with Genetic Algorithms. Proceedings of the 36th (IEEE) Hawaii International Conference on System Sciences, Waikoloa, Hawaii.

© 2016 IJAIR. All Rights Reserved