

Novel Point of View for Re-Engineering and Reverse Engineering through Genetic Algorithm in Software Testing

T. Rubini¹ M.Rajalakshmi²

¹M.Phil Student ²Assistant Professor

^{1,2}Department of Computer Science Engineering

^{1,2}Sankara College of Science and Commerce, Coimbatore

Abstract— Software testing is a process used to identify the accuracy completeness and quality of the well developed software. It includes a set of activities, conducted with the intent of finding bugs or errors in software so it corrected before product released to the client or end users. Software testing is an activity to check whether actual results match the expected results and to ensure that software system is error free or defect free. Software testing is more than just error detection. Testing software is operating the software under controlled conditions, to verify that it behaves as specified, finding bugs/errors and validate that what has been specified is what the user’s requirements. Actually testing is the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item. All of the tools using after generated software and subsequently undertaking initial test merely. In this research paper, we have discussed concerning how can test by means of infinity or else maximum set of values using with the assistance of genetic algorithm.

Key words: Software Testing, Tools, Genetic algorithm, Database, MTDT

I. INTRODUCTION

Grey box testing techniques combined the testing methodology of white box and black box. This approach implementing with proposed tool of Model Testing Database Tool (MTDT). Grey box testing technique is used for testing a piece of software against its specifications but using some knowledge of its internal working as well. Grey box testing includes reverse engineering to determine, for instance, boundary values or error messages. At this point new tool generated meant for testing (MTDT), this generate set of N number of data in a data table as per the database structure with same field name as well as data type. Genetic algorithm is been using for generating auto data insertion. Through, this Software Testing Engineers can be able to test their application with vast data and can find performance too.

II. CURRENT SYSTEM

Software tester works with the new software and help in repairing the errors occurred in it whereas Software developers are the professionals who develop the new software products depending on the client requirements. Software developer’s responsibilities also include broader aspects in development of software products. Software development comes with developer, software publisher, programmer and system development life cycle. Software testing is done by software testers, manager, testing lead, designer, software tester, automation developer and the test administrator.

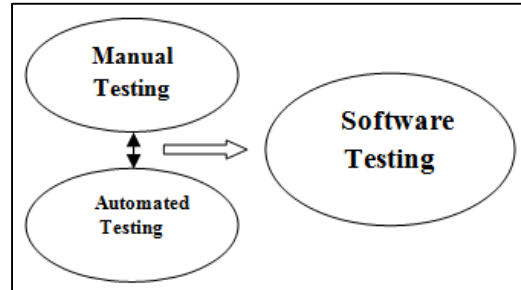


Fig. 1: Base of Software Testing

III. PROPOSED SYSTEM

Software development used to refer to the activity of computer programming, which is the process of writing and maintaining the software source code, but in a broader sense of the term it includes all that is involved between the conception of the desired software through to the final manifestation of the software, ideally in a planned and structured process. Therefore, software development may include research, novel development, model prototyping, modification, re-use, re-engineering, maintenance or other activities that result in software products Software Developer has to deal with all actions from the initial design up to the final testing of software.

Test cases: The basic objective of writing test cases is to validate the testing coverage of the application. Writing test cases brings some sort of standardization and minimizes the ad-hoc approach in testing.

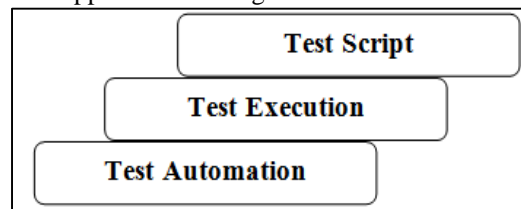


Fig. 2: Flow of Testing

Grey Box Testing	In the Gray box testing tester is usually has knowledge of limited access (LA) of code and based on knowledge, the test cases are designed and the software (S/W) application under test treat as a black box and tester test the application from outer. The name is comes because the application for tester is like a gray box like a transparent box and tester notice inside it but not fully transparent and can notice partially. To test the Web Services application usually the Gray box testing is used.
Black-box testing	Black box testing is a method of software testing (ST) that examines the functionality of an application without peering into internal structures. This method of test can be

	practical to virtually every level of software testing: The unit testing, integration-test, system testing and acceptance testing. Black box testing treats the system as a black box , so it does not explicitly use Knowledge of the internal structure or source code.
White box testing	This testing is a testing technique, that examine the programming structure and derives test data from the program logics or source code. The other names of glass box testing are clear-box testing, open-box testing, logic-driven testing or path driven testing or structural testing.

Table 1:

IV. GENETIC ALGORITHM

A. Operators of GA:

From the genetic algorithm outline, the crossover and mutation are the most important part of the genetic algorithm.

Start	Generate random population of N number of chromosomes.
Fitness	Evaluate the fitness value of $f(x)$, each chromosome "x" in the population.
New population	Selection: Select two parent chromosomes from a population according to their fitness (the better-fitness, the better chance to be selected) Crossover: With a crossover probability cross over the parents to form a new offspring. If no crossover was performed, offspring is an exact copy of parents. Mutation: With a mutation probability mutate new offspring at each locus. Accepting: Place new offspring in a new population.
Replace	Use novel generated population intended for an additional run of algorithm.
Test	If the end condition is satisfied, stop the process and return to the first best solution in current population.

Table 2: Outline of the Basic Genetic Algorithm

B. Encoding of a Chromosome:

The chromosome should in some way contain information about solution which it represents. The most used way of encoding is a binary string. The chromosome then could look like this:

Chromosome 1	1101100100110110
Chromosome 2	1101111000011110

Table 3:

Each chromosome has one binary string. Each bit in this string can represent some characteristic of the solution. The whole string can represent a number this has been used in the basic Genetic Algorithm (GA). There are many other ways of encoding. This depends mainly on the solved problem.

C. Crossover:

After decided what encoding will use, make a step to crossover. Crossover selects genes from parent

chromosomes and creates a new offspring. The simplest way how to do this is to choose randomly some crossover point and everything before this point copy from a first parent and then everything after a crossover point copy from the second parent.

Chromosome 1	11011 00100110110
Chromosome 2	11011 11000011110
Offspring 1	11011 11000011110
Offspring 2	11011 00100110110

Table 4:

There are other ways how to make crossover, for example can choose more crossover points. Crossover can be rather complicated and very depends on encoding of the encoding of chromosome. Specific crossover made for a specific problem can improve performance of the genetic algorithm.

D. Mutation:

After a crossover is performed, mutation takes place. This is to prevent falling all solutions in population into a local optimum of solved problem. Mutation changes randomly the new offspring. For binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. Mutation can then be following:

Original offspring 1	1101111000011110
Original offspring 2	1101100100110110
Mutated offspring 1	1100111000011110
Mutated offspring 2	1101101100110110

Table 5:

The mutation depends on the encoding as well as the crossover. For example when we are encoding permutations, mutation could be exchanging two genes.

V. MODEL TESTING DATABASE TOOL

The process or phase of creating software is called software development, once the software has been formed and ready to provide for end users, test the software by the requirements. Software testing is an execution of a program with the intention of finding bug. Development is writing the source code, testing is finding out whether or not the source code runs the way expect it to. Software testing is an inspection performed to present information about the quality of a product or software in test to the concerned users or clients. The trained professionals who perform these testing jobs are called software testers. Software testing includes many different techniques and is not only limited to execution of programs to find errors or software bugs.

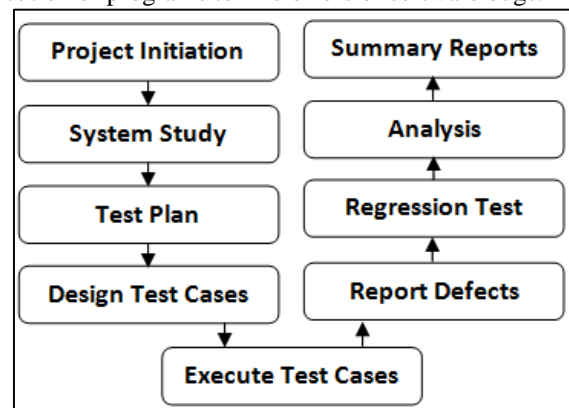


Fig. 3: Work flow of Execution of test cases

Generating chromosomes meant for all data types and inserting in database. That will generate thousands of data. So, execution part testing, performance testing, report generation can apply with full trust worthy flow.

VI. CONCLUSION

Our research tool MTDT is a common one to all the software applications that may be windows application or web applications. Everybody can easily check their application speed test and performance before delivery of the software. With the long term approach, this research is been taken.

REFERENCES

- [1] Myers, G.J. 1979. The Art of Software Testing. John Wiley & Sons, New York, NY, USA.
- [2] C.Wohlin, R. Prikladniki, 2013. Editorial: Systematic literature reviews in software engineering. Information and Software Technology.
- [3] M. Smit, E. Stroulia, 2013. Simulating service-oriented systems: A survey and the services aware simulation framework. IEEE Transactions on Services Computing.
- [4] Q. Taylor and C. Giraud-Carrier, "Applications of data mining in software engineering", Int. J. Data Analysis Techniques and Strategies, 2010.
- [5] A.-P. Li, Z.-h. Wang, L.-G. Duan and X.-P. Li, "Study and application of legacy system reengineering based on component reuse", Journal of Applied Sciences, vol. 13, no. 8, (2013).
- [6] Aman Jatain and Deepti Gaur, Reengineering Techniques for Object Oriented Legacy Systems, International Journal of Software Engineering and Its Applications Vol. 9, No. 1 (2015).