International Journal of Engineering Sciences Paradigms and Researches, Vol. 01, Issue 01, Oct 2012
ISSN (Online): 2319-6564
www.ijesonline.com

Software Testing Prioritization Based on Requirement Using Analytic Hierarchy Process

Ritu[1] and Dr. Nasib Singh Gill[2]

[1]Department of Computer Science and Applications, M. D. University, Rohtak-124001, Haryana, India
ritudhankhar08@gmail.com

[2]Professor, Department of Computer Science and Applications, M. D. University, Rohtak-124001, Haryana, India
nasibsgill@gmail.com

Abstract

Software Testing is a process to check the system/software is work properly as per expectation or not. Testing in software may be done at the completion of project or at some intermediate stages. It can be stated as process of validating and verifying that a software program/application meets the requirements and works as expected. To test the complete software is not possible hence we perform exhaustive testing. The testing domain of a software is very large but in exhaustive testing we run a set of test cases that can cover the maximum range of the software. Hence There is a need to select those test cases that can cover the maximum code and branches of software and return the maximum bugs. To select the high priority test cases, there is a need to improve our test cases selecting process. AHP stands for 'Analytic Hierarchy Process' is a systematic decision making method that is fully based on mathematical calculation in which we take pair wise comparison matrix having each unique pair of criteria. We have implemented the technique AHP to prioritize the testing criteria. To select the test cases, we used the previous experience of the employees that works under the testing experience. We performed a survey to collect the data about these testing criteria in various organizations. Gathered data is filtered and then we prepare a pair wise comparison matrix table by the data collected from survey. Using that table we can find the priority vector for various testing criteria used. And with the help of this priority vector, we can set the priority of the testing criteria. After prioritizing criteria we invest cost and time as per requirement, which are giving good results.

Keywords: *Software Testing, CFT, DFT.*

1. INTRODUCTION:-

Software is basically made up of codes and instructions that perform a special task for which we design it. Software testing is the process to verify that the product or the program we have made works as the expected and fulfils the requirements giving optimum performance. Software testing, depends on the testing method used, can be implemented at any phase in the development process. Generally, we perform the testing after all the requirements have been defined and the process of coding has been completed. It is less expensive to fix the bug or error in the starting phase of the development process.

There are mainly two **type of testing;** first one is static testing which includes dry run and code review. In dry run, we do not use computer to run program, we take any value of variable and check it on the paper thoroughly. And in code review, code is checked line by line. Second one is dynamic testing which includes white box testing and black box testing. White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these. This allows the software team to examine the parts of a system that are rarely tested and ensure that the most important function points have been tested [2]. While black box testing treats the software as a "black-box" without any knowledge of internal implementation. In this specification based testing aims to test the functionality of software according to the applicable requirements [3]. Specification based testing is necessary, but it is insufficient to guard against the risks [4].The black-box tester has no "bonds" with the code, and a tester's perception is that a code *must* have the bugs. By following the rule- Ask and you will receive, black-box testers find the bugs where programmers do not find. In other way, black-box testing has been said to be "like a walk in a dark room without a flashlight," because the tester doesn't know that how the software being tested was actually constructed. As a result, there are two circumstances when (1) a tester writes many test cases to check something that could have been tested with only one test case, and (2) some parts of the back-end are not tested at all. There are also some other type of testing comes under black box testing and white box testing. Concentrate

1.1*Control flow testing*:- control flow testing technique is applicable in the most of the software programs and application. It is very much effective for them. Control flow testing technique comes under the models of the black-box testing. But there is also some disadvantage in this technique in this testing we do not able to detect the any unwanted

feature that is already include in the software but n't a part of our requirement. It is mostly applied to the small programs or to the some part of the large programs and application.[5]

1.2 *Data flow testing:*-Data flow testing technique comes under the white box testing. In this testing, test cases are designed based on to the flow of the data with in the using codes. It is testing to test the flow of data in the application.[6] One problem that comes out with the guiding the testing of the module to meet the requirement of the test cases that were computed by using the results of the inter procedural data flow analysis.

1.3 *Regression Testing:*-Regression testing is basically to perform the testing again after any changes occur in the program [7]. It is because of, to check whether any change in one part of the program does not create any fault in the other part of the program. Or you can say that when we remove the bugs from the software, some changes occur[8]. We need to check that these changes should not create any new fault in the software. That is why the most of the software developing situation, it is to be assumed good coding practice that when a bug is found out the bug, is applied regularly after every change occurs in that program [9].

1.4 *Random Testing :-* It is also known as the 'Gorilla Testing'. In this technique, we do not test the software or the program completely and in the proper sequence. We select any part or any module from the program and done testing on it to check that whether it is performing well or there is some kind of scope of improvement in it. It comes under the black box testing design technique.

1.5 *Mutation Testing :*-In mutation testing, we select the set of mutation operators and then we implies them to the program one at a time for each of the source code that is applicable. It was proposed by Richard Lipton in 1971 [10]. Mutation testing modify the program in some small ways [11]. To verify the correctness of the software's implementation tests can be created.

1.6 *Functional Testing :*-Functional Testing is to test in a program to check a particular function how does that particular function perform.

1.7 *Structural Testing :*-It is also known as Glass Box testing and Clear Box testing. In this testing, we test the software with in the knowledge of the coding in the software and the internal structure of that software.

Therefore, black box testing has the advantages of "an unaffiliated opinion" and the disadvantage of "blind exploring"[12].

2. TECHNIQUE OF EVALUATION

WE USE "AHP"**:-**The AHP was introduced by Saaty[6] in 1980. The AHP is systematic decision making method that has been adapted for the software's requirements prioritization [13 14]. It is structured technique used for handling the complex decisions with the help of the psychology and the mathematics. It is used throughout the world in business, companies, hospitals, education etc. when complex decisions are taken. It is conducted by comparing all possible and unique pair of requirements to determine which of the two is of higher priority to what extend the total number of comparison n*(n-1)/2 (where n is number of requirements) are required to perform by the decision maker. In AHP the result is exponential increase in number of comparison as the number of requirements. Studies have shown that AHP is not suitable for large number of requirements[15 16]. AHP is vey trustworthy, in original form the redundancy of pairwise comparison allows a consistency check where judgments errors can be identified.

2.1 *Purpose :*-The main purpose of it to make the decision more effective and qualitative and to organizing the thoughts of the people. One of the importance to achieve the target and second of the performance in each of the criteria that are used.

2.2 *Procedure :-*

- To create the weight for the criteria
- To create the ranking for every decision for each criterion
- Find the average of all the decision alternative and choose the highest one.

More advantages is that the resulting priorities and based on a ratio scale which allows for the useful assessments of requirements Saaty [13] states that intensity of importance should be according to table 1.

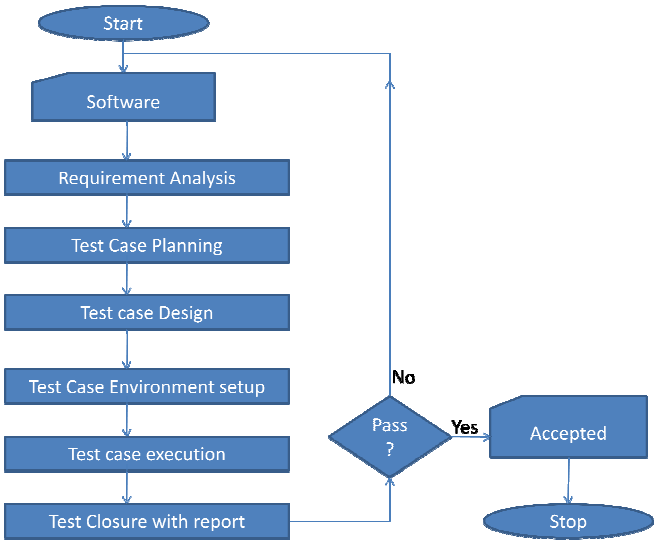
Table 1. Fundamentals scale for pair wise comparisons

Intensity of importance	Description
1	Of equal importance
3	Moderate difference in importance
5	Essential difference in importance
7	Major difference in importance
9	Extreme difference in importance
Reciprocals	If requirement i has one of the above numbers assigned to it when compared with requirement

	j , then j has the reciprocal value when compared with i

3. TESTING CRITERIA:-

According to the software testing life cycle we take six basic criteria for software testing



As we can see in the diagram the software has to pass all these criteria and only in that case, the software will be accepted otherwise we have to repeat the whole process again. To understand about all these phases, the brief overview is given here.

3.1 *Requirement analysis(RA) :*-As per SRS(software requirement specification) document, we select user requirements and analyze them.In this phase, testing team reads the requirements from a testing point of view to identify the testable requirements. The Testing team may interact with various stakeholders likeClient, Business Analyst, Technical Leads, System Architects to understand the requirements in detail. Requirements may be either Functional or Non Functional .Various activities in requirement analysis are to identify types of tests to be performed, to set the testing priorities and focus, prepare Requirement Traceability Matrix

3.2 *Test planning(TP)* :-After analyzing the requirements, we plan to make test cases which can mostly cover/test the software. In this process, the project manager, test manager make meeting for the planning.In this phase the Project Manager has to decide about the things which have to be tested and what is the appropriate budget needed for this. The proper planning at this stage would greatly reduce the risk of the low quality software. Proper planning in this stage would include preparation of the high level test plan. It is a process how the test process should follow. This phase is also known as strategy phase. in this stage, a Senior Tester will determine effort and cost estimates for the project and would prepare the Test Plan. Various activities are performed in this phase such as Preparation of test plan for various types of testing, to test the tool selection, the test effort estimation, resource planning.

3.3 *Test Cases Development(TCD) :*-In this phase the test cases will be developed in actual. The test manager and his team make test cases as per test case planning.In this phase the creation, verification and rework of test cases & test scripts is done. Test cases are identified/created and are reviewed and then reworked as well. Various activities which are performed in this phase are to create test cases, to review and baseline the test cases.

3.4 *Test Environment Setup(TES) :*-As per the hardware and software requirement in SRS, we setup test environment. Test environment is used to decide the software and hardware conditions under which software product is tested. The establishment of Test environment is one of the critical aspects of testing process and can be done in parallel with Test Case Development Stage**.** Testing team may not be involved in this activity if the customer/development team provides the test environment in which case the test team is required to do testing of the given environment.Various activities performed in this phase are to study the required architecture, environment set-up and prepare hardware and software requirement list for the Test Environment and to setup test Environment and test data.

3.5 *Test Execution(TE) :*-In this we will execute the test cases and make the log of results and their exhibition time.SRS, if it satisfactory, we pass the software for release otherwise send back to the development team. The test cases are executed and defects are reported in bug tracking tool, after the test execution is complete and all the defects are reported. Testexecution reports is sent to project stakeholders. After that developers fix the bugs raised by testers they give newer build with fixes to testers, testers perform re-testing and regression testing to ensure that the defect has been fixed and not affected any other areas of software.After tester assures that defects have been fixed then we will go for the next phase.Various activities in this test execution phase are to execute test cases as per plan, to document test case's results, and log the defects for failed test cases, to map the defects to test cases in RTM, to retest the defect fixes and to track the defects to closure .

3.6 *Test Cycle Closure (TCC):-* We will observe the report with SRS. If it is satisfactory, we pass the software for release otherwise send back to the development team. All the members of Testing team will meet , discuss and analyze testing artifacts to identify strategies that have to be implemented in future, taking lessons from the current test cycle. The basic idea is to remove the process bottlenecks for future test cycles and share best practices for any similar projects in future. Various activities performed in this phase are to evaluate cycle completion criteria based on budget, Test coverage, Time Software Critical Business Objectives ,quality etc, to prepare test metrics based on the given parameters, to document the learning out of the project , to finalize Test closure report, to qualitative and quantitative reporting of quality of the work product to the customer, to test result analysis to find out the defect distribution by type and severity.

4. ANALYSIS AND RESULTS:-

Using the fundamentals in table 1, we perform a survey in the company(Beryl System company Faridabad). After survey we get the following results for all these testing criteria that are shown in table 2. By using this table we perform further tasks those will help us in setting the priorities of the testing criteria. Here is the initial table from Beryl System company Faridabad survey, we get pair wise matrix.

	RA	TP	TCD	TES	TE	TCC
RA	1	1/3	1/5	1/7	1/3	1/5
TP	3	1	1/3	1/5	1/7	9
TCD	5	3	1	1/7	1/3	9
TES	7	5	7	1	9	5
TE	3	7	3	1/9	1	5
TCC	5	1/9	1/9	1/5	1/5	1

Table 2 – Pairwise Comparison Matrix

Now we normalize the above table by using the process in which firstly we perform the addition of each column separately and then divide each value in table by the addition valueof that particular column. After normalization of the pair wise comparison matrix we get table 3.

	RA	TP	TCD	TES	TE	TCC
RA	0.04	0.02	0.01	0.07	0.03	0.006
TP	0.12	0.06	0.02	0.11	0.01	0.30
TCD	0.20	0.18	0.08	0.07	0.03	0.30
TES	0.29	0.30	0.60	0.55	0.81	0.17
TE	0.12	0.42	0.25	0.06	0.09	0.17
TCC	0.20	0.006	0.009	0.11	0.01	0.03

Table 3- after normalization

Now we calculate the priority vector(PV) using table 3.For finding the value of priority vector we perform the row-wise addition that means the value of priority vector for each testing criteria will be the sum of all the elements in that particular row.

	RA	TP	TCD	TES	TE	TCC	PV
RA	0.04	0.02	0.01	0.07	0.03	0.006	0.176
TP	0.12	0.06	0.02	0.11	0.01	0.30	0.62
TCD	0.20	0.18	0.08	0.07	0.03	0.30	0.86
TES	0.29	0.30	0.60	0.55	0.81	0.17	2.72
TE	0.12	0.42	0.25	0.06	0.09	0.17	1.11
TCC	0.20	0.006	0.009	0.11	0.01	0.03	0.365

Table 4 – Normalized Table having PV

In this table we find priority vector by the formula given by Eigen. On the basis of the values calculated for priority vector, we will assign the priority of these various testing criteria.

Highest priority vector value is 2.72 by TES hence we will give the highest importance to TES after that TE got priority vector value 1.11 it will get second highest priority. At position third TCS comes with the priority vector 0.86 and then on number fourth there is TP with priority vector 0.62. and on second last position TCC with the value of priority vector value 0.365 and at below of all RA with the priority vector value 0.176. the exact order we got is TES>TE>TCD>TP>TCC>RA.

5. CONCLUSION:-

From the above discussion and the calculation that are performed on the importance of software testing,we can conclude that, to perform a good testing there should also be good environment of testing should not be that run the test cases there should be a proper planning for the testing. As above the order is shown TES>TE>TCD>TP>TCC>RA. We can also consider the factors in the priority order to make the test cases. If this order will be followed then the design test cases will give the better results.

REFERENCES:

[1] Exploratory Testing, CemKaner Florida Institute of Technology, *Quality Assurance Institute Worldwide Annual Software Testing Conference*, Orlando, FL, November 2006
[2] Introduction, Code Coverage Analysis, Steve Cornett
[3] Laycock, G. T. (1993) (PostScript). *The Theory and Practice of Specification Based Software Testing*. Dept of Computer Science, Sheffield University, UK.Retrieved 2008-02-13.
[4] Bach, James (June 1999). "Risk and Requirements-Based Testing" (PDF).*Computer***32** (6): 113–114. Retrieved 2008-08-19.
[5] www.softwaretestinggenius.com

[6] P. G. Frankl, S. N. Weiss, and E. J. Weyuker,"ASSET: A system to select and evaluate tests,"Proceedings of the IEEE Conference on Software Tools, New York, April 1985.
[7] Myers, Glenford (2004). *The Art of Software Testing*.Wiley.ISBN 978-0-471-46912-4
[8] Savenkov, Roman (2008). *How to Become a Software Tester*.Roman Savenkov Consulting.p. 386.ISBN 978-0-615-23372-7
[9] Kolawa, Adam; Huizinga, Dorota (2007). *Automated Defect Prevention: Best Practices in Software Management*.Wiley-IEEE Computer Society Press.p. 73.ISBN 0-470-04212-5
[10] Mutation 2000: Uniting the Orthogonal by A. Jefferson Offuttand Roland H. Untch.
[11] A Practical System for Mutation Testing: Help for the Common Programmer by A. Jefferson Offutt
[12] Savenkov, Roman (2008). *How to Become a Software Tester*.Roman Savenkov Consulting.p. 159.ISBN 978-0-615-23372-7.
[13] T.L. Saaty, The Analytic Hierarchy Process, McGraw-Hill, Inc. (1980).
[14] Regnell B, Host M, Nattoch Dag J, Beremark P, HjelmT(2001),"An industrial case study on distributed prioritization in market driven requirements engineering for packaged software. Requirements Engineering 6(1):51-62.
[15] Lehtola :, Kauppinen M(2004) Empirical Evaluation of Two Requirements Prioritization Methods in Product Development projects, Proceedings of the European Software Process Improvement Conference(EuroSPI 2004), Springer-Verlag, Berlin Heidelberg, pp. 161-170.
[16] Maiden NAM, NcubeC(1998) Acquiring COTS Software Selection Requirements. IEEE Software 15(2):46-56.