A Single Fromat for Measuring different Aspects of Testing

Mahwish Kundi
Department of ICIT
Gomal University
Dera Ismail Khan, Pakistan

Shahid Hussain
Department of Computing
Namal College
Mianwali, Pakistan

Bashir Ahmad
Department of ICIT
Gomal Univeristy
Dera Ismail Khan,Pakistan

Muhammad Inayat Ullah
Department of CS & IT
NWFP, University of Engineering
& Technology, Peshawar, Pakistan

Abstract—— **In-Process testing metrics has been used from some years and its usage is frequently increasing. There are different metrics for software testing i.e to measure testing progress, Mean time between arrival of error, density of errors, fixation of errors, failure rate, test execution Productivity, cost of defects, Test efficiency and efficiency checking and so on. But all these metrics are independent and have no relation with each other. There are some attributes of these metric which are very much homogenous and interrelated with interdisciplinary measurement. It is quit natural to inter-relate all these metrics into a single metric which should depict overall functionality of some of existing selected metrics. So the derived frame work modeled a new metric. This new metric covers the measurement of major quality attributes such as Correctness, Reliability, Efficiency, Usability and Maintainability. The derived new metric possess higher level of reliability, early predication of testing progress, process less cost of correctness, effectiveness in error exploration, efficient approach of testing, Compatibility of different existing metrics, less cost of corrective maintenance, high degree of flexibility and interoperability of different Tools.**

Keywords- Metrics, Software Testing, Testing Metrics, In- Process Testing Metrics, Quality Attributes

I. I. INTRODUCTION

The software development process is completed via different phase's i.e Analysis, Requirement specifications, Design, Coding, Testing and Maintenance. The software testing is one of the most important phases of software development life cycle. Software testing has three main purposes: verification, validation, and defect finding [1]. This phase has a direct impact on end product. Testing phase is the most expensive and lengthy phase. It can never be certain that a testing system is correct [2].

The metrics are used to measure the software i.e software metric is a measure of some property of a piece of software or its specifications [3]. The different metrics are used to measure the different phases of software in order to determine the progress of software development process.

In similar way there is a number of metrics used for measuring testing process of software. Test metrics accomplish in analyzing the current level of maturity in testing and give a projection on how to go about testing activities by allowing us to set goals and predict future trends [4]. The different testing metrics are used independently to covers different aspects of software testing process.

Some of these testing metrics are as below

Cost of finding a defect in testing (CFDT):Test Case Adequacy: Test Case Effectiveness: Effort Variance : Schedule Variance: Schedule Slippage: Rework Effort Ratio: Review Effort Ratio: Requirements Stability Index: Requirements Creep: Weighted Defect Density:[5].

II. TESTING AND MEASURMENTS

A. Testing

The IEEE definition of testing is: "Testing is the process of exercising or evaluating a system of system component by manual or automated means to verify that it satisfies specified requirements". Testing is one of the SDLC most important phases. In this phase the error are removed from software to provide the high quality product to customer.

B. Test Case

A test case is a set of the some rules or checks which is fixed for the testing of a system. Test Cases is generated by tester mostly, some time also by those programmer which performs the testing role also. In test case consist of input, precondition of execution and output of a particular component.

C. Metrics

Metrics means measurement. Metrics are very important in the development of software as it helps to identify the errors in the software development and thus helps to ensure the quality of the software. [6].The types of software metrics are Project Metric, Process Metric and Product Metric [7].

D. Testing Metrics

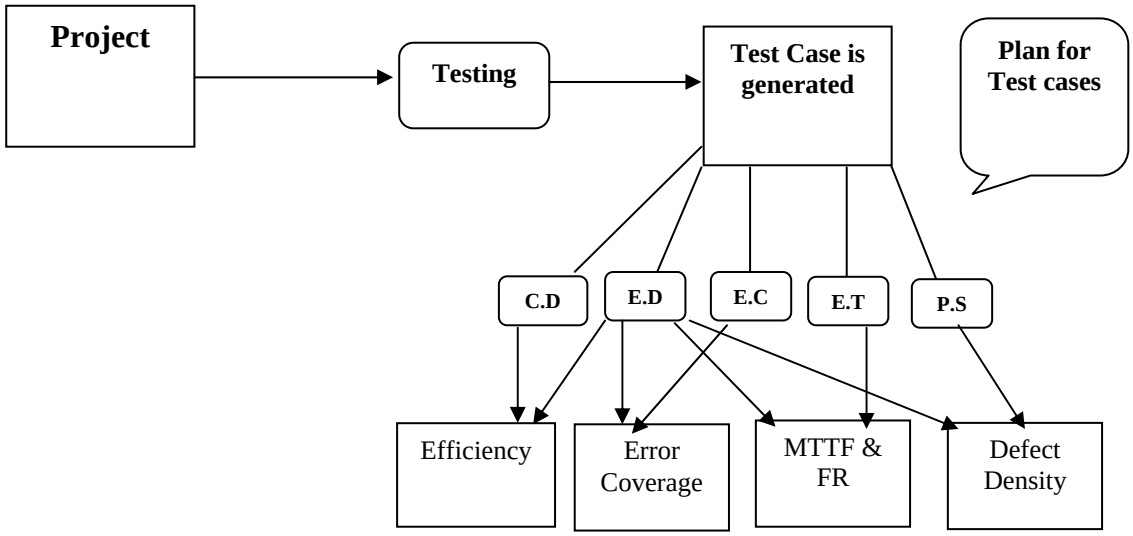
Like the measurement of other phases the measurement of testing phase is also required. A number of software testing metrics are available. Some of the major testing metrics are, Delivered defect quantities, Product volatility, Defect ratios, Defect removal efficiency, Test coverage, Cost of defects, Costs of quality activities [8].

E. In-Process testing metrics

In-process tracking and measurements play a critical role in software development, particularly for software testing [9]. S-curve, Testing Defect Arrivals over Time, Testing Defect Backlog over Time, Product Size over Time, CPU Utilization During Test, System Crashes and Hangs, Mean Time to Unplanned IPL [10].

II. A SINGL MODEL FOR DIFFERENT MEASURMENT OF TESTING

The different testing metrics are integrated into a single framework to measure some important aspects of testing process. The steps for the new metrics are discussed in detail as under.



Figer1. Frame Work for Measuring Different Aspects of Testing

The above model shows the steps as at first a project or system is taken to perform testing on it. Then the test cases are generated and then proper planning is done for each test case. Then the further steps are interrelated with each other which provide the different measurements of testing process of selected project. The parameter required for selected metrics are abbreviate as C.D, E.D, E.C, E.T, P.S in above model. Where C.D stands for Detected Error on Customer side, E.D stands for Error Detection, E.C for Error Correction, E.T stands for Execution Time taken by each test case, P.S stands for Program size tested.

The formulas for above model is as below

$$MTTF = \frac{\text{Execution Time}}{\text{No. of Error}}$$

$$\text{Failure Rate} = \frac{\text{No. of Error}}{\text{Execution Time}}$$

$$\text{Error Coverage} = \frac{\text{No. of Corrected Error}}{\text{No. of Error}} * 100$$

$$\text{Defect Density} = \frac{\text{No. of Errors}}{\text{Program Size Tested}} * 100$$

$$\text{Efficiency} = \frac{\text{No. of Error}}{\text{No. of Error + Customer Side Error}} * 100$$

IV. IMPLEMENTATION FOR FRAME WORK OF DERIVED METRIC.

A.Measurment for Testing Plan

Through S-curve the proper plan for testing will be determined and at early stage the plan for testing can be changed if the testing process is falling behind. As the time will spend the test cases planning should be also increase.

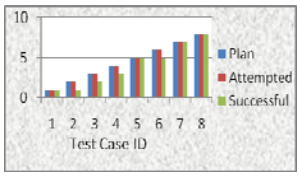


Figure-2 Plan for Testing

B.Measurment of error correction and detection efficeincy

The next step is to find the errors in planned test cases and also do make correction of them. So from this step there is a greater chance of exploring errors, because of planed testing process. This step will reduce the corrective maintenance cost, because the error correction efficiency is measurable at this early stage. So this error detection and correction efficiency can be improved at this early stage.

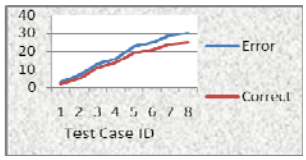


Figure-3 Correction & Detection

C. Measurment of Execution time for error correction and detection

The next step in proposed model is to measure the execution time takes by each error. From this execution time the severity of error is known, as the error takes more time in correction or exploring is more sever. This step cleared that test case with greater number doesn't mean more that it consumed more time and more costly.

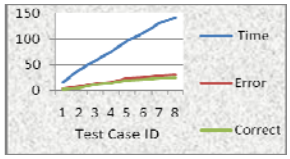


Figure-3 Execution Time

D.Measurment of errors in programe size

The program size will be increased as the number of test cases will be increased and from calculating program size the defect density of testing process can be measured. From where the more buggy line of code will be determined. So from this step the more defective portion of programs or buggier component of a program is determined.

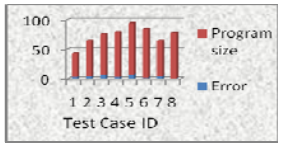


Figure-4 Defect Density

E. Measurement for Reliability

For measuring the reliability the MTTF or MTBF is used. Reliability is the ability of a system or component to perform its required functions under stated conditions for a specified period of time [11].. The higher the MTBF number is, the higher the reliability of the product [11] so the high value of MTTF shows the more reliable test case. Because if the error will appear after long duration then the time to failure will increase i.e error will occur after long time and system will perform its intended operation for long time. So from this step the reliability of a system is determined.

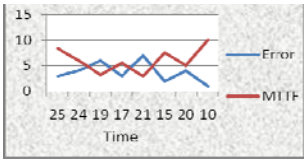


Figure-5 MTTF

F. Failure Rate

Failure rate is the frequency with which an engineered system or component fails, expressed for example in failures per hour [12]. This is also included in derived new metric for measurement of reliability, because the failure rate is inverse of MTTF. Failure rate will increase if MTTF will decrease. The higher value of failure rate shows the low reliability of system, because the occurrence rate of error will be increased. So the greater rate of failure causes of lower reliability i.e low value of mean time to failure.

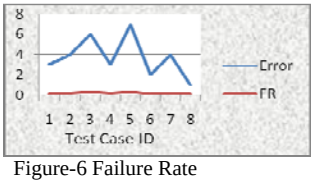


Figure-6 Failure Rate

G. Measurement for testing Efficiency

This step provides the measurement of efficiency of testing team or tester. is measured to determine As the efficiency is used to measure the efficiency of the test team and also individual Tester. Efficiency is the No of bugs found / No. of bugs found on user side [13].

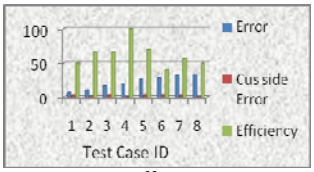


Figure-7 Efficiency

Table 1A single Model Contain Different Measurements of Testing

Test Case ID	Error	Error Corrected	Customer Side Error	Execution Time in Min	Program Size	Error Coverage in %	Defect Density	MTTF	Failure Rate	Efficiency in %
1	3	2	3	25	40	66.67	7.5	8.33	0.12	50
2	4	3	2	24	60	75	6.67	6.0	0.17	66.67
3	6	6	3	19	70	100	8.57	3.17	0.32	66.67
4	3	3	0	17	76	100	3.95	5.67	0.18	100
5	7	5	3	21	87	71.43	8.05	3.0	0.34	70
6	2	2	3	15	82	100	2.44	7.5	0.13	40
7	4	3	3	20	60	75	6.67	5.0	0.2	57.14
8	1	1	1	10	77	100	1.29	10.0	.1	50

This derived metric provide a new combined approach for measurement in which a proper frame work is based on some In-process testing metrics and also included some other metric which covers some basic attributes of software quality i.e correctness, Reliability, Efficiency, Usability and Maintenance.

Must keep tabs on what we had planned and what we have actually achieved for measuring efficiency [14]. So it is cleared that different tester focuses on different factors for measuring testing efficiency in software testing, this metric provides a combined approach for measuring the major factors effecting testing process such as Proper Plan, Error Detection and Error Correction, Execution time, Program size tested, Defect Density, Error Coverage, MTTF and Failure Rate etc.

V. CONCLUSION

This new frame work provides a proper way of measuring different aspects of testing process. There is a number of software testing metrics such as Failure Rate, MTTF, MTBF, Defect Density, Test Plan, Testing Efficiency, Error fixation etc. These metrics are independent and have no link with each other, but some these have homogeneous attributes with interdisciplinary measurement. Some of these homogeneous attributes metrics has been integrated into single frame work. So from this single frame work some major measurement of testing process such that plan for testing, error detection and correction efficiency, program size tested, testing efficiency, execution time take by each test case, the reliability measurement using MTTF and FR can be performed. This derived frame provides the measurement of software testing process at different stages, from which the effect of each step taken in software testing is determined. The derived metric also provides the higher degree of flexibility where different metrics are combined in a single frame work to be used more effectively and also provides the compatibility of different metrics. The term plan testing is also included in new derived metric which is helpful in exploring more errors, and this reduces the corrective maintenance cost. From this derived metric more reliable and efficient product can be achieved with low cost of maintenance. It has a direct impact on product quality because this new metric provides the measurement of some basic attributes of quality such that Correctness, Reliability, Efficiency, Usability and Maintainability. This new metric is applied on different tools such that VB.Net and SQL and provides the interoperability of different tools.

REFRRENCES

[1].John E. Bentley, Wachovia Bank and Charlotte NC , “Software Testing Fundamentals—Concepts, Roles, and Terminology
[2]. Andrew Ireland, “The Software Testing Life-Cycle”
[3] http://en.wikipedia.org/wiki/Software_metric
[4]. http://www.onestoptesting.com/test-metrics/
[5].http://www.softwaretestingstuff.com/2007/10/software-testing metrics.html
[6]. http://qacampus.com/blog/?p=49
[7]. Sanjay B. Dange, “Test Metrics in Iterative Model of Development”, Datamatics Limited Nashik, Maharashtra, India, 2003.
[8] http://www.exforsys.com/tutorials/testing/metrics-used-in-testing.html
[9].S.H.Kan, J.Parrish and D.Monlove “In-process metrics for software testing” ,2001
[10] Stephen H. Kan,“Metrics and Models in Software Quality Engineering”, Second Edition, Addison-Wesley Professional, 2002.
[11].Wendy Torell and Victor Avelar, “Mean Time Between.Failure: Explanation and Standards” American Power Conversion, 2004.
[12]. http://www.r2launch.nl/index.php/failure-analysis/failure-rate
[13]. http://www.geekinterview.com/question_details/35870.
[14].http://www.ezilon.com/articles/articles/4583/1/Practical Measurements-For-Software-Testing