A Comparative study of Test Exit Criteria in Software Testing

Ishita Verma
House No.4, Village Dayalpur, Karawal Nagar Road
Delhi-110094, India

ABSTRACT

Software Testing refers to the set of activities carried out on the software with the intent of finding errors. It is one to ascertain that the software meets all its specifications. For any software a very large number of scenarios and input value combinations exist and testing the software for each of these combinations is not possible. And, also the time and resources for testing are limited so the challenge of testing is to test maximum functionality in the available time period. Hence, Test Exit criteria or when to stop testing, is of utmost importance. In this paper we will discuss the different Test Exit criteria and focus on their comparison to determine which one of them is the best which covers maximum functionality of the software**.**

General Terms

Software Testing

Keywords

Testing process, Test exit criteria, Test closure activities, Post -test activities

1. INTRODUCTION

Software testing is the set of activities carried out on the software with the intent of finding errors [5]. Software testing is carried out to ascertain that the software meets all its specifications. Effective software testing ensures delivery of a better quality software, low maintenance costs and more satisfied users whereas ineffective software testing leads to the delivery of a low quality product [3].

Data used for testing is the Test Data [1]. Testers analyze the requirements and design the test cases. Test case is a scenario containing the steps to follow, test data, execution conditions and the expected result. During testing, we follow the steps of the test case and obtain the Actual result [3]. The Actual result is compared with the Expected result. If the two match then the test case passes otherwise the test case fails. Errors lead to Fault in the Programs. Error may be human error like a developer has not been taught some development practice or he has not understood the requirements properly. When the program having a fault executes, then it results in a Failure. So, Fault is a passive entity whereas Failure is an Active entity [2]. Failure is the deviation of a program's output from the Expected output. Fault is also called as Bug or Defect.

A collection of test cases is called as a Test Suite (see Fig. 1) For any software a large number of scenarios and a large number of input values exist and testing each and every scenario and value is not feasible hence, exhaustive testing is not possible. The requirements or the specifications which serve as the basis for testing, and using which the test cases are designed, is called as Test Basis.

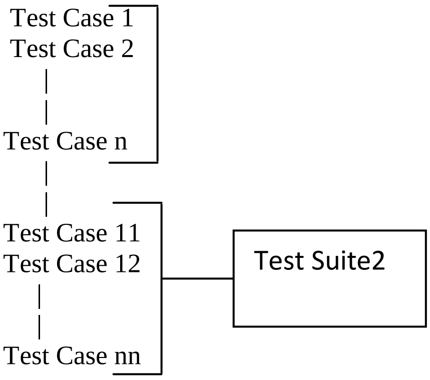

Fig 1. Test Case and Test Suite

2. AN OVERVIEW OF SOFTWARE TESTING PROCESS

The different phases of Software Testing Life Cycle, as shown in Fig. 2 are as follows:

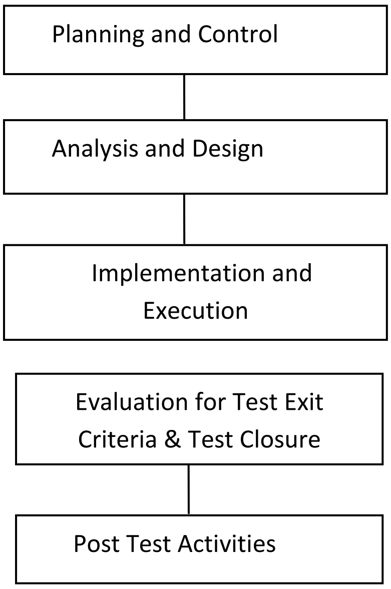
Planning and Control
Analysis and Design
Implementation and Execution
Evaluation for Test Exit Criteria & Test Closure
Post Test Activities

Fig. 2: Software Testing Life Cycle

2.1 Planning and Control

Planning is an Umbrella activity [5] and is done for all the activities and sub activities carried out during the testing life cycle. Planning means, deciding on the approach to be followed for obtaining the test objective. In this Phase, Test plan is made which consists of all the Functionalities to be followed, Effort required, Time, Cost, the number of Test cases to be written etc. Control means regulations and monitoring that the things are going as per the plan and making modifications as and when needed.

2.2 Analysis and Design

Analysis means analyzing the different requirements to obtain the scenarios and sub scenarios which are to be tested. Design means conversion of the scenarios and sub scenarios identified in Analysis phase into logical Test cases.

2.3 Implementation and Execution

Implementation refers to the conversion of Logical test cases into Physical test cases. Execution means executing the Test cases on the software and recording the Result obtained. Whenever the Actual and Expected result of a Test case do not match then an Incident occurs which needs to be investigated [4]. If the Incident is a Bug then the Bug needs to be recorded in a Defect Report in the proper format with all the necessary information.

2.4 Evaluation for Test Exit Criteria & Test Closure Activities

Depending on the criteria mentioned in the Test plan about, when to stop Testing, the Exit criteria for Testing are determined [3]. It could be (1) 80% Test cases executed with Pass status or, (2) 40 Requirements out of which 20 are of High priority have been tested or, (3) Coverage of functionality / Requirements reaches a specified point. So, depending upon the Test exit Criteria the decision to stop testing is taken by the Test Managers. Test Closure Activities means we will identify the deliverables like Test Summary, Test reports, Test Results etc. to be given to the customer and they all are constructed. The bugs in the Defect Report are closed after they have been fixed.

3. POST TEST ACTIVITIES

Post Test Activities are also called as Post Mortem Activities [6]. It involves recording how many test cases were executed, how many bugs were found? These metrics tell how Productive testing was? These metrics are used for future estimation of similar projects. These also suggest all the improvements that can be made to improve testing.

4. METHODOLOGY

The methodology used to study and compare different Text Exit Criteria is given below.

4.1 University Result Management System

This is the software selected for study and comparing different Test Exit Criteria. This software is used for managing the result of Students in a University. The different modules that it contains are:

- Student information module: This contains all the information of a student who is admitted to the university.
- Subject information module: This contains all the information about different subject combinations available.
- Result module: This contains information about the marks secured by students in different subjects, their total score, percentage and corresponding grades.
- Co-curricular activities module: This contains information about all the co-curricular activities of a student along with Descriptive Indicators for the student.
- Grade Card generation module: This module generates mark sheet for a student containing all his personal information, subject information, marks in different subjects and Descriptive Indicators for him.

4.2 High Priority Functional Requirements

Some of the High Priority User Requirements for University Result management system are:

- The software should not allow unauthorized access.
- The system should display result term wise for each of the three terms and also final result.
- The system should display result subject wise.
- The system should display result for a student when his roll No. or name is entered.
- The system should also generate the student list for a particular class.
- The system should allow saving and printing the result.
- The system should generate Grade card for a student including all his information.
- The system should allow saving and printing of the grade card.
- The Grade card should contain marks in individual subjects, total marks (Sum of the marks obtained in all the subjects), percentage and the corresponding grade for the student.

Based on the above functional requirements, different scenarios are identified and then those scenarios are converted into Test Cases.

4.3 Matrix showing Functional Requirement ID and corresponding Test case ID

A simple Matrix showing the relationship between functional requirements and the Test cases is as shown in Table 1 below:

Table 1: Matrix showing the relationship between functional requirements and the Test cases

Functional Requirement ID	Test Case ID
FRS_01	TC_01, TC_02, TC_03
FRS_02	TC_04, TC_05, TC_06, TC_07, TC_08, TC_09, TC_10
FRS_03	TC_11, TC_12, TC_13, TC_14, TC_15, TC_16, TC_17, TC_18, TC_19
FRS_04	TC_20, TC_21, TC_22, TC_23

FRS_05	TC_24, TC_25, TC_26, TC_27, TC_28, TC_29, TC_30, TC_31, TC_32, TC_33
FRS_06	TC_34, TC_35, TC_36, TC_37, TC_38, TC_39, TC_40, TC_41, TC_42, TC_43,
FRS_07	TC_44, TC_45, TC_46, TC_47, TC_48, TC_49, TC_50, TC_51, TC_52, TC_53
FRS_08	TC_54, TC_55, TC_56, TC_57, TC_58, TC_59, TC_60, TC_61, TC_62, TC_63
FRS_09	TC-64, TC_65, TC_66, TC_67, TC_68, TC_69, TC_70, TC_71

4.4 Different Test Exit Criteria

The test cases are executed and testing is exited using different Test Exit Criteria as follows:

4.4.1 80% of the high priority requirements have been tested

Total High Priority Requirements are 9 so, 80% of that = 7, hence when 7 requirements have been tested then testing will be exited. This means when Test cases corresponding to FRS_01, FRS_02,……FRS_09 have been tested then Testing is exited.

So, total High priority requirements tested here =7.

4.4.2 25 High priority bugs have been found

Now, Test Summary Report is as follows:

Total number of test cases run = 40 (TC_01, TC_02,…TC_40)

Number of test cases failed = 25

Total number of fully tested functional requirements = 5 (FRS_01, FRS_02, FRS_03, FRS_04, FRS_05)

4.4.3 80% of the Test cases have been executed

Total number of test cases = 71

So, 80% of that = 56

Now, Test summary report after testing 56 Test cases is shown in Table 2.

Table 2: Test summary report of 56 Test cases

Test Case ID	Corresponding FRS
TC_24, TC_25, TC_26, TC_27, TC_28, TC_29, TC_30, TC-31, TC_32, TC_33	FRS_05
TC_11, TC_12, TC_13, TC_14, TC_15, TC_16, TC_17, TC_18, TC-19	FRS_03
TC_34, TC_35, TC_36, TC_37, TC_38, TC_39, TC_40, TC_41, TC_42, TC_43	FRS_06
TC_44, TC_45, TC_46, TC_47, TC_48, TC_49, TC_50, TC_51, TC_52, TC_53	FRS_07
TC-54, TC_55, TC_56, TC_57, TC_58, TC_59, TC_60, TC_61, TC_62, TC_63	FRS_08
TC_64, TC_65, TC_66, TC_67, TC_68, TC_69, TC_70, TC_71	FRS_09

So, total number of fully tested High priority requirements = 6

4.4.4 Comparison of the different Test Exit Criteria:

Comparing the above mentioned three Test Exit Criteria, we obtain the following results shown in Table 3.

Table 3: Comparison of three Test Exit Criteria

Test Exit Criteria	**Number of Fully tested High Priority requirements**
80% of High Priority requirements have been tested	07
25 high priority bugs have been found	05
80% of the Test Cases have been executed	06

5. CONCLUSION

As is clear from the above comparison table, Test Exit Criteria should always be based on the number of High priority requirements that have been tested as that results in the maximum number of requirements being tested. All the other criteria which are based on the number of high priority bugs found or the number of test case executed may not actually test different functional requirements because there is a possibility that if a particular requirement has been misunderstood and implemented wrongly then there is possibility that all the test cases which have been derived using that requirement may have bugs. So, the best Test exit criterion is the one based on the number of High Priority requirements that have been tested.

6. ACKNOWLEDGMENTS

The help rendered by friends and colleagues in finalizing this paper is duly acknowledged.

7. REFERENCES

[1] Boris Beizer, 1990, “Software Testing Techniques”, Second Volume, Second Edition, Van Nostrand Reinhold, New York.

[2] Louis Tamres, 2002 “Software Testing”, Pearson Education Asia,

[3] R.S. Pressman, 2012, “Software Engineering – A Practitioner’s Approach, 6th Edition, Mc. Graw Hill Int Edition..

[4] Ram Chillarege, 1999, “Software testing Best Practices”, IBM Research

[5] K.K. Aggarwal & Yogesh Singh, 2005, “Software Engineering”, 2nd Edition, New Age

[6] Ian Somerville, 2006, Software Engineering, 8TH edition