A Study on the Efficiency of a Test Analysis Method Utilizing Test-Categories Based on AUT and Fault Knowledge

Tsuyoshi Yumoto, Keiji Uetsuki, Toru Matsuodani, and Kazuhiko Tsuda

Abstract—Along with a rapid increase in the size and complexity of software today, the number of required test cases is also increasing. Generally, a rather significant number of testers should be assigned to a project in order to manage this increase of test cases. It is necessary for such a large team to be able to collaborate and work together efficiently, and we have proposed a test analysis method for black box testing utilizing Test-Categories based on Application Under Test (AUT) and fault knowledge in order to facilitate this activity. This paper illustrates the effectiveness of Test-Categories.

Keywords—Software testing, test analysis, test condition, Test-Categories

I. INTRODUCTION

Along with a rapid increase in the size and complexity of software today, the number of required test cases is also increasing. In the case of black box testing, the number of required test cases is equal to the function point sum to the power of 1.3 to 1.5 [1]. Sizes in the function point of projects have grown nearly 10 times from 1970 to 2000 [2]. There is also a survey the volume of embedded software is increasing at ten to twenty percent per year depending on the domain [3]. Generally, a significant number of testers should be assigned to a project in order to manage this increase of test cases. There is a case that approximately 5,000 people were assigned to a project for testing over a period of 8 months in Japan [4]. Although the majority of testers are involved in the activity of test development, there are no clearly defined general rules for test development. Rather, they are developed according to the individual’s own judgment. This has the potential to cause the lacking or duplication of test cases. This also can cause deterioration in the test cases [5]. To help solve this issue, a set of rules for a test analysis method for black box testing utilizing Test-Categories based on the Application Under Test (AUT) and fault knowledge have been proposed [6]. This paper illustrates the effectiveness of the Test-Categories test analysis method.

II. VARIABILITY OF TEST ANALYSIS RESULTS

In the ISTQB, the test development process is defined as consisting of test analysis, test design, and test implementation [7]. This study has focused on test analysis in the test development process. Test analysis is described as “…during test analysis, the test basis documentation is analyzed in order to determine what to test, i.e., to identify the test conditions [7].” Although necessity of and requirements for performing the test analysis are defined, the approach for analyzing the test basis remains undefined in ISTQB, or in several studies by Ostrand [8], Grindal [9], etc. Moreover, previous studies have focused on the techniques for test parameter design which are applied after test analysis in which test conditions determined, and test conditions considered on the assumption that these are already prepared [8]-[11], etc.

When the rules are not clearly defined, the output of test analysis is classified by each individual’s own judgment. Because the test conditions, the output of test analysis, include many things, the analysis results can vary by individual [6].

The great variability of test analysis results can be observed in Fig.1, which show results of experiments conducted to verify and measure the effectiveness of the Test-Categories method. The details of the experiments are described in section IV. Some typical results from the experiments are shown in Fig.1.
1) CS#1 and CS#2: Input fields of the AUT are listed in the row titles. However, the contents in the table and the way in which column names are different.

2) CS#3: It cannot be determined which contents are related to which actions, parameters, and expected results because each column is independent of the others.

3) CS#4: States of the AUT are listed as the column names. Parameters and values which are possible in each state are written in the table.

Fig. 1. Test analysis results when rules are not defined and interpretation is left to the individual.

This shows that different individuals can reach different results. The different results mean a great variance in test conditions determined through test analysis. Thus, when the test analysis is performed by many individuals there is a high chance for test conditions to either be duplicated or even missed out entirely. Some studies on the test analysis method have also been proposed by Nishi [12], Akiyama [13], etc. While these studies look at the logic of analyzing AUT, they have little focus on how effective these methods actually are in reducing duplication and missing test conditions when working with more than one person.

III. AN APPROACH OF TEST CATEGORIES BASED TESTING

In the case of black box testing, the internal structure of the AUT cannot fully be known, and it is necessary to execute testing only via input and output.

Omura states "... the artificial systems, because it is a converter that adds value to convert the input into the output, logically, has a structure like Fig. 2 [14]." The proposed method has used a similar concept. We have considered each feature to be tested [15] to be an artificial system with a similar logical structure. The logical structure can be used to test the feature in a Mutually Exclusive and Completely Exhaustive (MECE) way [16], such as in Fig.3. Each box in Fig.3 can be a useful guide to determine the required test conditions.

![Fig. 3. A MECE way to determine test conditions from features.](Image)

We can posit the hypothesis that by implementing a set of rules it will be easier to determine the necessary test conditions to ensure the appropriate testing takes place. The following issues currently make this difficult:

1) Certain aspects of specification are not written if they are thought to be obvious.

2) Specification is not completely written within single target section in a document. (for example: a behavior about a combination of functions)}
conditions, including specification-items, expected results, and test parameters, from each feature. A specification-item (spec-item) is one functional requirement from a Functional Design Specification document that describes the behavior of a single feature within the AUT (for example: "The volume can be set between 1 and 10, where 1 is silent and 10 is 100dBs"). A test parameter is a pre-condition or a pre-input or a combination of pre-conditions and pre-inputs for a test case [17].

The most important thing is to build consensus on the decided Test-Categories. Therefore, in order to ensure that each member clearly understands the meaning of each Test-Category the members must discuss examples and potential failures and/or faults which may arise for each Test-Category.

### TABLE I

<table>
<thead>
<tr>
<th>Logical Structure</th>
<th>Test-Category</th>
<th>Example</th>
<th>Potential Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conversion</td>
<td>Calculation</td>
<td>Fee calculation</td>
<td>Calculation error resulting from inability to handle decimals.</td>
</tr>
<tr>
<td>Input adjustment</td>
<td>UI input</td>
<td>Validation rules to the input form. Screen control</td>
<td>The same airport as a departure and an arrival will be able to be input to book one way ticket.</td>
</tr>
<tr>
<td>Management</td>
<td>Reflection</td>
<td>Data handling reflected on other functions</td>
<td>Errors due to referring to data which has been deleted.</td>
</tr>
</tbody>
</table>

Test-Categories discussion is recorded in the table such as the one shown in Table I. Members involved in the test development activity are able to build consensus regarding Test-Categories. The results of this consensus building include:

1) All of the testers were able to attain a similar level of understanding with respect to the AUT  
2) Variation in the interpretation of the test conditions amongst the testers was minimized

Moreover, in the proposed method, it has been mentioned that the steps used to determine test conditions from a test basis are executed in the following procedure [6].

1) Step1: Select feature to be tested  
2) Step2: Design Test-Categories  
3) Step3: Select and organize specification-items and expected results using Test-Categories  
4) Step4: Select and organize test parameters

When determining a test condition from a test basis, a test condition list such as the one shown in Table II is used. The same set of Test-Categories is listed for all features to be tested, and the specification-items and expected results corresponding to each Test-Category are listed. Depending on the features to be tested, specification-items relevant to Test-Categories may not be listed in the case where they are not applicable. Such a case should be indicated as Not Applicable (N/A). In the case of certain specification-items, the expected results can vary depending on various conditions. Thus, it is necessary to list multiple expected results for these specification-items.

<table>
<thead>
<tr>
<th>Features to be tested</th>
<th>Test-Category</th>
<th>Spec-item</th>
<th>Expected result</th>
<th>Test parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-a</td>
<td>TC-a</td>
<td>SI-a</td>
<td>ER-a</td>
<td>TP1,TP2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SI-b</td>
<td>ER-b-1</td>
<td>TP1,TP3,TP4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ER-b-2</td>
<td>TP1,TP4</td>
</tr>
<tr>
<td></td>
<td>TC-b</td>
<td>SI-c</td>
<td>ER-c</td>
<td>TP5,TP6,TP7</td>
</tr>
<tr>
<td></td>
<td>TC-a</td>
<td>SI-d</td>
<td>ER-d</td>
<td>TP9,TP10</td>
</tr>
<tr>
<td></td>
<td>TC-b</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Through creating a test condition list, the combination of test parameters required for each set of specification-item and the expected result will be able to be determined. These test parameters will be designed equivalent classes and combinations in the test design process. As we mentioned in section I, there are several studies focusing on methods or techniques for test parameters design.

When many testers are involved in test development and proceed according to these steps, all of the testers can carry out their work according to the same set of rules. As a result, the developed suite of test conditions are more comprehensive and do not contain duplicates. This ensures higher test coverage overall, delivering higher quality testing. This is the main benefit of this method.

Furthermore, there are three benefits to this procedure:

1) The procedure is based on the structure of the test conditions defined in the proposal of this method [6]. Elements which are included in the test basis are classified into specification-items, expected results, and test parameters. One-by-one, these are determined and selected through this procedure. Thus, the end result readability is improved since each member’s results derived from the test analysis were created with the same logic and procedures, and can be arranged in the same classification item categories.
2) Once a consensus regarding the building of Test-Categories is met, team members can more easily determine and select specification-items.
3) Since this method is structured, standardized and easy to walk though, testers will come up with consistent and repeatable test analysis.

Verification experiments were performed to prove the effectiveness of this prevention.
IV. REMARKS FROM THE VERIFICATION EXPERIMENT

A. Procedure of the verification experiment

The verification experiments were conducted during a workshop. In this workshop it was explained that a test development process is required in order to derive test cases. After showing the test basis for a particular exercise, the test analysis was carried out. The participants were grouped into teams of four or five members. The results derived from the first exercise, where the analysis methods are not predetermined, are shown in Fig.1. It can be seen that the each team came up with a different results, because they were analyzed their individual own approach. After all participants recognized these results, the test analysis method was explained, and test analysis exercise was conducted again according to the specified procedure.

The verification experiments were conducted twice. For the first, the exercise was carried out for a music reproduction equipment as the AUT (6 teams), and for the second, a flight booking application as the AUT (2 teams).

The total time required for the workshop was 4 hours.

B. Evaluation of the verification experiment

The specification-items results of the exercise conducted according to Test-Categories and the results of the exercise conducted without knowing the Test-Categories method were compared. Fig.4 is the one comparison result table from the first verification experiment. Test analysis results for the exercise conducted without knowing this method have variation as shown by Fig.1. Therefore, in order to count the number of specification-items, the facilitator has arranged the answers of each team in Fig.4 form after the completion of the workshop.

<table>
<thead>
<tr>
<th>Feature</th>
<th>NOT knowing the Test-Categories</th>
<th>Knowing the Test-Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvA</td>
<td>N/A 1 1 1 1</td>
<td>N/A 1 1 1 1</td>
</tr>
<tr>
<td>ConvB</td>
<td>N/A 0 0 0 0</td>
<td>N/A 0 0 0 0</td>
</tr>
<tr>
<td>Input</td>
<td>N/A 0 0 0 0</td>
<td>N/A 0 0 0 0</td>
</tr>
<tr>
<td>ConvA</td>
<td>N/A 1 1 1 1</td>
<td>N/A 1 1 1 1</td>
</tr>
<tr>
<td>ConvB</td>
<td>N/A 0 0 0 0</td>
<td>N/A 0 0 0 0</td>
</tr>
<tr>
<td>SupportA</td>
<td>N/A 0 0 0 0</td>
<td>N/A 0 0 0 0</td>
</tr>
<tr>
<td>SupportB</td>
<td>N/A 0 0 0 0</td>
<td>N/A 0 0 0 0</td>
</tr>
<tr>
<td>OutputA</td>
<td>N/A 0 0 0 0</td>
<td>N/A 0 0 0 0</td>
</tr>
<tr>
<td>Storage</td>
<td>N/A 0 0 0 0</td>
<td>N/A 0 0 0 0</td>
</tr>
<tr>
<td>MngtA</td>
<td>N/A 0 0 0 0</td>
<td>N/A 0 0 0 0</td>
</tr>
</tbody>
</table>

Fig. 4. A comparison result table from team2(TM2) of the first verification experiment.

As we can see by considering these evaluations, there was a measurable improvement resulting from implementing the Test-Categories method for seven out of eight teams. According to these evaluations, observations for the each verification experiment were below:

1) In the case of the first exercise, where the AUT was music reproduction equipment, volume control was the feature to be tested. The number of specification-items listed within each category increased for five teams. When comparing the logical structure boxes that sum up the Test-Categories, there were five teams whose number of listings increased in Management. In the case of Output and Storage, since the complete specification-items were already listed in the first exercise it can be said that the number of listings of increased for all teams for which at increase was possible.

2) In the case of the second exercise, where the AUT was a flight booking application, a new flight booking registration was the feature to be tested. The number of specification-items listed within each category increased for both teams. When comparing the logical structure that sum up the Test-Categories, Conversion, Output, and Storage increased for both teams.

In this study, eight comparison results table were taken from the two verification experiments. These comparison result tables were gathered to one table to evaluate effectiveness of the method. When sum up these comparison result tables to one table, the evaluation levels shown in Table III were used to make the evaluation. The experimental results are shown in Fig.5.

<table>
<thead>
<tr>
<th>Evaluation level</th>
<th>Comparison Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Number of listed specification-item did not increase, and is less than suggested answer.</td>
</tr>
<tr>
<td>A</td>
<td>Number of listed specification-item did increase, and is less than the suggested answer.</td>
</tr>
<tr>
<td>A+</td>
<td>Number of listed specification-item did increase, and manages to be the same as the suggested answer.</td>
</tr>
</tbody>
</table>

As we can see by considering these evaluations, there was a measurable improvement resulting from implementing the Test-Categories method for seven out of eight teams. According to these evaluations, observations for the each verification experiment were below:

1) In the case of the first exercise, where the AUT was music reproduction equipment, volume control was the feature to be tested. The number of specification-items listed within each category increased for five teams. When comparing the logical structure boxes that sum up the Test-Categories, there were five teams whose number of listings increased in Management. In the case of Output and Storage, since the complete specification-items were already listed in the first exercise it can be said that the number of listings of increased for all teams for which at increase was possible.

2) In the case of the second exercise, where the AUT was a flight booking application, a new flight booking registration was the feature to be tested. The number of specification-items listed within each category increased for both teams. When comparing the logical structure that sum up the Test-Categories, Conversion, Output, and Storage increased for both teams.

Fig. 5. The evaluation result of the two verification results.
the method. Therefore, based on the results of the verification experiments, we have further analyzed using patterns of input and output data of the AUT.

C. Evaluation using patterns of input and output data of AUT

For an executed test, data is input into the AUT, and the AUT output is confirmed by comparing the actual output with the expected output. For example, when verifying whether a calculation result of a simple function, such as executing an addition or subtraction operation, is correct, one would input values from the outside of the AUT into the AUT, the AUT would calculate these values, and then the AUT would output the result of the calculation to the outside of the AUT. This is an example of Data-in from outside, Data-out to outside, as shown in Fig.6. When a fixed rate is applied to the calculation result, (meaning, another calculation is applied), the calculation result is output from the AUT only after calling and using the appropriate rate. This is an example of Data-in from outside & inside, Data-out to outside, which is also shown in Fig.6.

![Diagram of AUT input-output patterns](image)

Fig. 6. An explanation of Input-data and Output-data of AUT.

The way in which data is input into the AUT can be classified into three patterns. Likewise, the way in which data is output from the AUT can also be classified into three patterns. Therefore, the total combined patterns of inputting and outputting data into/from the AUT (I/O data patterns) can be summarized into the nine patterns in Fig.7.

![Diagram of I/O data patterns](image)

Fig. 7. An explanation of the I/O data patterns.

It should be noted that the I/O data pattern is observable from the outside of the AUT. Data used for starting/ending (such as event or signal) is not considered when determining classification. This is because this method is a black box testing analysis method, and internal commands such as events or signals in the AUT are not considered for this level of testing.

The Fault Model proposed by Whittaker [18] is a somewhat analogous study about modeling the input and output patterns of the AUT. The objective of the Fault Model is to find faults. In the proposed method, the AUT model is used to classify specification-items, which are determined from the test basis. In order to classify the results of the verification experiments into the nine I/O data patterns, a designation of P1 to P9 is added as an attribute to the specification-items. These are organized based on the approach shown in Table III.

The experimental results are shown in Fig.8. The number of rows differs in Fig. 5 and Fig.8 because some specification-items in the same Test-Category are added different I/O data patterns and the Test-Categories are not summarized in the logical structure. According to Fig.8, P1, P2, P4, and P7 correspond to the AUT used in the verification experiments.

![Table of I/O data patterns](image)

The summarized results of the I/O data pattern are shown in Fig.9. The percentage of combined scores of A and A* for each input-output pattern have been recorded. P2 is 100% for both the experiments. It can be said that P2 has had the most significant result.

![Table of evaluation results](image)

Fig. 8. The evaluation results of the I/O data pattern.

The summarized results of the I/O data pattern are shown in Fig.9. The percentage of combined scores of A and A* for each input-output pattern have been recorded. P2 is 100% for both the experiments. It can be said that P2 has had the most significant result.

![Table of evaluation results](image)

Fig. 9. The summarized results of the I/O data patterns.

In the case of the music reproduction equipment, the specification-item classified into P2 was the preservation of a volume pre-set value. Although this specification-item was described in the test basis, it was indicated somewhere other the section describing volume control. In the case of the flight booking application, the specification-item classified into P2...
was the order details saved in a database. There was no detailed description about inserting a booking data into a database in the test basis. These observations may suggest that the results are aligned with our assumption, certain aspects of specification are not written since it is thought to be obvious, which was asserted in Section III.

V. CONCLUSION

Through these verification experiments it has been observed that after briefly explaining this proposed method to participants there was a measurable improvement in quantity and consistency of spec-item which they were able to determine. Moreover, by analyzing I/O data patterns, the patterns of inputting and outputting data into/from the AUT, a part of the evaluation results from the verification experiments aligned with the assumption in the proposed method. Further verification experiments are necessary in order to carry out trend analysis with higher accuracy. Conducting further experiments and deepening our understanding of the tendencies and factors relating to effectiveness of the proposed method, rules for creating Test-Categories based on the AUT knowledge and fault knowledge can be more refined.

REFERENCES