A Study of Boundary Value Analysis and Equivalence Class partitioning in Software Testing

Mr. BHARAT B. DAREKAR

Department of Computer Science, KTHM College, Nashik

ABSTRACT

Software testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is_defect free. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements._Boundary Value Analysis (BVA) is a black box software testing technique where test cases are designed using boundary values. BVA is based on the single fault assumption, also known as critical fault assumption which states that failures are rarely the product of two or more simultaneous faults. Equivalence Partitioning or Equivalence Class Partitioning is type of black box testing technique which can be applied to all levels of software testing like unit, integration, system, etc. In this technique, input data units are divided into equivalent partitions that can be used to derive test cases which reduce time required for testing because of small number of test cases.

INTRODUCTION

Software testing or rather exhaustive software testing is a very time and resource-intensive activity. In order to effectively test any application in the best possible time and with optimal resources, we use different test design techniques. One such technique is boundary value analysis. Boundary value analysis is a black-box testing technique. It is closely associated with equivalence class partitioning. In this technique, we analyze the behavior of the application with test data residing at the boundary values of the equivalence classes. The main concept behind boundary value testing is to choose input variable values that are –minimum, just above minimum, just below the minimum, nominal value, just below maximum, maximum, and just above maximum. That is, there are two borders for each range: the lower boundary (the start of the range) and the higher boundary (the end of the range), and the boundaries represent the start and end of each valid division. We should create test cases that put the program's functioning to the test at its limits, using values just within and beyond the limits. Stress and negative testing both include boundary value analysis.

The technique of testing between extreme ends or borders between divisions of input data is known as boundary testing.

• These extreme endpoints, such as Start-End, Lower-Upper, Maximum-Minimum, Just Inside-Just Outside values, are referred to as boundary values, and boundary is used to test them.

- In normal boundary value testing, the main concept is to choose input variable values that are
 - o Minimum, Just a little over the bare minimum
 - A minimal amount of money, Just below the upper limit
- Equivalence Class Partitioning is useful in boundary testing.
- After Equivalence Class Partitioning, follows Boundary Testing.
- It splits program input data into several equivalent data types.
- When there is a range in the input field, you may use this strategy.

Why Equivalence & Boundary Analysis Testing

- 1. This testing is used to reduce a very large number of test cases to manageable chunks.
- 2. Very clear guidelines on test cases without compromising on the effectiveness of testing.
- 3. Appropriate for calculation-intensive applications with a large number of variables/inputs

Equivalence Partitioning is divided into two sections

1. Pressman Rule – If the input consists of a range of numbers, create test cases for one valid and two incorrect values. If the input consists of a collection of values, provide test cases for all legal value sets as well as two incorrect values.

Consider the following scenario -

Consider any online shopping website, where each product should be identified by a unique ID and name. Users may search for products using either the product name or the product ID. You may look at a list of items with product IDs and see whether any of them are Laptops (valid value).

Create test cases for both true and false values if the input is Boolean. Consider the following example web page, which has text fields for first name, last name, and email address, as well as radio buttons for gender that employ Boolean inputs.

The relevant value should be set as the input if the user clicks on any of the radio buttons. If the user selects a different choice, the input value must be changed to reflect the new selection (and the previously selected option should be deselected).

When a radio button option is selected, it is considered as TRUE, and when none is selected, it is treated as FALSE. Furthermore, two radio buttons should not be chosen at the same time; if they do, this is considered a problem.

2. Method of Practice – Divide the range of values into comparable portions for all of the valid values while also ensuring that two incorrect values are tested.

Advantages of Boundary Value Analysis

1. It is easier and faster to find defects using this technique. This is because the density of defects at boundaries is more.

2. Instead of testing will all set of test data, we only pick the one at the boundaries So, the overall test execution time reduces.

Disadvantages of boundary value analysis

- The success of the testing using this technique depends on the equivalence classes identified, which further depends on the expertise of the tester and his knowledge of the application. Hence, incorrect identification of equivalence classes leads to incorrect boundary value testing.
- 2. Applications with open boundaries or applications not having one-dimensional boundaries are not suitable for this technique. In those cases, other black-box techniques like "Domain Analysis" are used.

METHODOLOGY

Example 1: Equivalence and Boundary Value

- Let's consider the behavior of Order Pizza Text Box Below
- While value 11 to 99 are considered invalid for order and an error message will appear, "Only 10 Pizza can be ordered"

Submit

Pizza orders

Here is the test condition

- 1. Any Number greater than 10 entered in the Order Pizza field (let say 11) is considered invalid.
- 2. Any Number less than 1 that is 0 or below, then it is considered invalid.
- 3. Numbers 1 to 10 are considered valid
- 4. Any 3 Digit Number say -100 is invalid



The divided sets are called Equivalence Partitions or Equivalence Classes. Then we pick only one value from each partition for testing. The hypothesis behind this technique is that if one condition/value in a partition passes all others will also pass. Likewise, if one condition in a partition fails, all other conditions in that partition will fail.



Boundary Value Analysis- in Boundary Value Analysis, you test boundaries between equivalence partitions



In our earlier equivalence partitioning example, instead of checking one value for each partition, you will check the values at the partitions like 0, 1, 10, 11 and so on. As you may observe, you test values at both valid and invalid boundaries. Boundary Value Analysis is also called range checking.

Equivalence partitioning and boundary value analysis (BVA) are closely related and can be used together at all levels of testing.

CONCLUSIONS:

When it is virtually hard to evaluate a huge pool of test cases separately, boundary analysis testing is performed. The methods employed include boundary value analysis and equivalence partitioning testing. First, you separate a collection of test conditions into a partition that may be examined in Equivalence Partitioning. The borders between equivalence divisions are next tested using

Boundary Value Analysis. Appropriate for applications that need a lot of calculations and include variables that reflect physical quantities.

REFERENCES:

- The Art of Software Testing, 3rd Edition-Glenford J. Myers, Corey Sandler, Tom Badgett.
- 2. Software Testing, 2nd Edition, 2005-Ron Patton.
- 3. How to Break Software: A Practical Guide to Testing- James Whittaker
- 4. Software Testing Techniques, 2nd edition- Boris Beizer