

Software Testing: Difference Between Equivalence Partitioning and BVA



Nikita Warma

Jan 4, 2017 · 3 min read

In this article about software testing, we are going to see the concepts of equivalence partitioning and boundary value analysis (BVA). These two are the very basic concepts in software testing, and are taught in various software testing class in Pune, conducted by various software testing institutions in Pune.

Let's introduce this concept to you, so that it would be easier to understand while pursuing a course in software testing.

To begin with, boundary value analysis and equivalence partitioning are test case design techniques that come under black box testing.

Test cases related to input box accepting numbers in the range of 1 and 1000 with the help of BVA.

1. BVA or Boundary Value Analysis:

It's generally perceived that input values at the upper and lower limit of input domain lead to more bugs in the system. More application blunders happen at the limits of input domain. 'BVA' Testing strategy is utilized to extract errors at limits, instead of finding those that exist at the centre of the input domain.

BVA is a next piece of Equivalence partitioning for creating test cases where test cases are chosen at the edges of the equivalence classes.

Here is an illustration of the same:

1. Test cases with test data precisely as the input limits of input domain i.e. values 1 and 1000 for our situation.
2. Test data with values just beneath the extraordinary edges of input areas i.e. values 0 and 999.

3. Test data with values simply over the limit edges of input space i.e. values 2 and 1001.

Boundary value analysis is frequently called as a piece of stress and negative testing.

Note: There is no firm rule to test just a single value from every equivalence class you made for the various input domains. You can choose different valid and invalid values from every equivalence class as indicated by your necessities and past judgments.

E.g. on the off chance that you separated 1 to 1000 input values in valid data equivalence class, then you can choose test case inputs like: 1, 11, 100, 950 and so forth. Same case for other test cases having invalid classes of data.

This ought to be an extremely essential and straightforward case to comprehend the Boundary value analysis and the concept of equivalence partitioning.

2. Equivalence Partitioning:

In this technique the input domain data is isolated into various equivalence data classes. This technique is normally used to lessen the aggregate number of test cases to a limited set of testable test cases, as yet covering maximum requirements.

In short it is the way towards taking all conceivable test cases and setting them into classes. One test value is chosen from every class while testing.

E.g.: If you are testing for an input box accepting numbers from 1 to 1000 then there is no utilization in writing 1000 test cases for every one of the 1000 legitimate input numbers in addition to other test cases for invalid information.

Making use of the equivalence partitioning strategy, above test cases can be partitioned into three arrangements of input data called as classes. Every test case is a representative of individual class.

So in above case we can isolate our test cases into 3 equivalence classes of some legitimate and invalid inputs.

Test cases for input box taking numbers somewhere around 1 and 1000 utilizing Equivalence Partitioning:

1) One input data class with all valid inputs. Pick a solitary value from range 1 to 1000 as a valid test case. In the event that you select different values somewhere around 1 and 1000 then outcome will be same. Therefore, one test case for valid data ought to be adequate.

2) Input data class with all values beneath lower limit. I.e. any number below 1, as an invalid input data test case.

3) Input data with any value more than 1000 to depict third invalid input class.

So utilizing equivalence partitioning you have classified all conceivable test cases into three classes. Test cases with different qualities from any class ought to give you a similar outcome.

We have chosen one representative from each input class to create the test cases. Test case values are chosen in a manner that biggest number of attributes of equivalence class can be worked out.

Equivalence partitioning utilizes least test cases to cover majority requirements.

For a successful career in software testing, undergoing a software testing course in Pune, is the initial step.

Testing Software Testing

[About](#) [Help](#) [Legal](#)