Boundary Value Analysis vs Equivalence Class Partitioning

Software testing is an essential aspect of software development that ensures that software products are functional, reliable, and meet the required specifications. Testing is crucial to identify software errors and bugs that can lead to system failures, data loss, and other negative impacts. Boundary Value Analysis (BVA) and Equivalence Class Partitioning (ECP) are two widely used testing techniques that help in Software Testing. In this article, we will discuss BVA and ECP, their benefits, real-life examples, and some common pitfalls to avoid.

What are Boundary Value Analysis and Equivalence Class Partitioning?

Boundary Value Analysis (BVA):

BVA is a testing technique that focuses on testing the boundary conditions of a system. The objective of BVA is to identify and test the values at the boundaries, which are often the most vulnerable points in a software application. These boundaries can include input limits, output limits, or any other type of boundary that is relevant to the system being tested.

The concept of BVA is based on the assumption that errors are more likely to occur at the boundaries of input values, rather than within the range of acceptable values. This is because the boundary values are usually handled differently by the software application, and any discrepancies in the handling of these values can result in errors.

For example, let’s consider a software application that accepts input values in the range of 1 to 100. The boundary values, in this case, would be 1, 100, and values close to them, such as 2, 99, etc. Testing the boundary values would involve checking how the software application behaves when it receives input values at the edge of the acceptable range. This includes testing values such as 0, 101, -1, 102, and so on.

Contrary to the name it’s less of an analysis but more of a heuristic that can be applied while testing, which means that it can fail too. It is important to note that BVA should not be used as the sole testing technique, but rather as a heuristic that can be applied in conjunction with other testing techniques.

Equivalence Class Partitioning (ECP):

ECP is an important technique used in software testing. At its core, ECP is about grouping similar items that have a common property into a set, where each set represents an equivalence class. Doing so, helps testers identify and target specific areas of the software that may have a higher likelihood of containing bugs or errors.

When you're driving a car, do you consciously calculate the exact amount of pressure needed to apply on the gas pedal to accelerate to a specific speed? No, you don't.

You know that pressing the pedal with a certain amount of force will generally get you to the desired speed. This is because the car's engine and transmission are designed to work together to achieve a range of speeds, so there is a range of acceptable inputs for achieving a given output. That's ECP! We apply it to anything we use regularly, like appliances, tools, and technology.

In the context of software testing, ECP is used to group tests based on various criteria such as input/output values, user groups, product risks, test environments, etc.

For instance, if you are testing a login page, you could group tests based on different input values such as valid usernames and passwords, invalid usernames, invalid passwords, and so on. Each group would represent an equivalence class that exposes different kinds of bugs or issues that may arise when users attempt to log in.

While ECP is an effective testing technique, it is not foolproof, and it can sometimes fail too. It is a thought process that relies heavily on the tester's knowledge and experience.

Boundary Value Analysis (BVA) and Equivalence Class Partitioning (ECP):

Now that we know about BVA and ECP, it is worth pointing out that they both are usually used together. Let's try to understand this with an example. Let's say you are trying to test a form that accepts age as a number:

– You cannot go on and create 100 test cases for it to test for all 100 inputs from the age of 1 to 100.

– You will also need to test the numbers below 1 and above 100. These are the boundary values.

– To make our testing more effective, let's employ ECP. Let's partition the 1 to 100 into groups of 10, so they would be [1-10];[11-20];[21-30];[31-40];[41-50];[51-60];[61-70];[71-80];[81-90];[91-100];

– For the created groups, the testing can be done for values on boundaries like: -1;0;1;9;10;11;19;20;21… and so forth

Benefits of Using BVA & ECP in Software Testing

While BVA and ECP are widely used, let's understand some of the key benefits of applying BVA and ECP in software testing:

1. Improved Understanding of the System Under Test: When we apply BVA and ECP techniques, we try to identify the boundaries of the system and group the input parameters into classes. This helps us to gain a better understanding of the system and its behavior.

2. Simplified Test Design: By grouping the tests based on input/output values, user groups, product risks, test environments, etc., testers can better plan their tests. This ensures that each class is tested comprehensively and exposes different kinds of bugs.

3. Better Test Coverage: BVA and ECP techniques help to identify critical tests by dividing the input domain into smaller regions, leading to effective test coverage. BVA focuses on testing boundary conditions, which are the most error-prone areas, while ECP divides the input domain into equivalence classes that can be tested as a group.

4. Better Prioritization: By using these techniques, we can prioritize the test cases based on their criticality. BVA helps in identifying the values at the boundaries, which are more likely to cause errors, while ECP helps in grouping the tests based on the common properties, which makes it easier to prioritize the tests.

5. Better Risk Management: BVA and ECP help to identify and manage risks associated with software testing. By identifying the most critical scenarios, testers can focus on the areas of the software that are most likely to cause problems.

Applications of BVA & ECP in Software Testing

The most common use/application of BVA and ECP in Software testing is found to be on the input values for the system under test. Apart from input values, the following are some of the other scenarios where these techniques can be applied:

Boundary Value Analysis (BVA):

1. Database Testing: In a database system, BVA can be used to test the boundary conditions of various parameters such as the number of records, data types, field sizes, the minimum and maximum length of character fields, the range of numeric fields, and the maximum number of records that can be stored, etc.

2. Network Testing: BVA can be used to analyze the traffic on the network and identify any abnormal conditions or boundary conditions that may cause the network to fail. In network communication protocols, BVA can be applied to test the boundary conditions of data packets, such as the minimum and maximum packet size, packet header length, and packet content. This technique can help ensure that the protocol will work correctly under a wide range of data transfer scenarios.

3. Hardware Testing: BVA can also be applied to hardware components, such as sensors, where boundary conditions like temperature, humidity, and pressure can impact the functionality of the component. By testing these limits, testers can ensure that the hardware component will function correctly under extreme conditions.

4. Testing of time-based functionality: BVA can be applied to test the time-based functionality of a system, such as a booking system, to check if it handles the boundaries of time zones, daylight saving time changes, and other related constraints.

5. UI Testing: BVA can be used to test the boundaries of the screen size, font size, color, alignment, and various user interface elements such as text boxes, buttons, drop-down menus, etc. By testing these limits, testers can ensure that the user interface will function correctly when users interact with it in unexpected ways.

Equivalence Class Partitioning (ECP):

1. Output values: Equivalence Class Partitioning can also be used to test output values. For example, if a software application is supposed to return a specific set of outputs for certain input values, you can use ECP to partition the expected output values into equivalent classes and test only one value from each class.
2. User roles: In software testing, ECP can be applied to different user roles. Different user roles may have access to different sets of features or functionality in an application. By applying ECP to user roles, you can identify and test the most critical combinations of user roles and features.
3. Time and date: ECP can also be used to test software applications that rely on time and date data. For example, if an application requires specific actions to be taken at specific times, ECP can be used to partition the relevant time periods into equivalent classes and test only one time from each class.
4. Environment: ECP can also be used to test software applications in different environments. For example, an application may behave differently on different operating systems or with different hardware configurations. By applying ECP to different environments, you can identify and test the most critical combinations of environment and software configurations.
5. Error Handling: ECP can also be used to test error messages. By partitioning error messages into equivalent classes, you can test only one message from each class, ensuring that all possible error messages are tested without the need to test every possible input value that might trigger an error.

Should you automate your tests for BVA and ECP Techniques?

You may be wondering whether or not to automate the tests for Boundary Value Analysis (BVA) and Equivalence Partitioning (ECP) techniques. In my opinion, both of these techniques involve testing a range of input values systematically, making them ideal candidates for automation.

Automating your tests for BVA and ECP can be a great way to save time and ensure accuracy in your testing. And with tools like Testsigma available, it’s easier than ever to automate your tests without the need for extensive coding or technical knowledge.

Automating your BVA and ECP tests with Testsigma can help you catch bugs and issues early on, before they cause major problems down the line. It can also help you achieve better test coverage by allowing you to run tests more frequently and consistently. Plus, Testsigma’s cloud-based nature means that you can access and run your tests from anywhere, making it easy to collaborate with remote team members or testers.

Automating tests for BVA and ECP can be highly beneficial for improving testing efficiency, consistency, and accuracy. However, it’s important to recognize that not all tests can be automated and that the decision to automate tests should be made after carefully considering the cost and benefit of automation.

Common Pitfalls to Avoid When Using BVA & ECP

When using Boundary Value Analysis (BVA) and Equivalence Partitioning (ECP) in software testing, it’s important to avoid certain pitfalls. Some common pitfalls to avoid include:

1. Restricting to only input values: BVA and ECP are not limited to input values alone. Testers must consider all possible factors that could affect the software, such as different user groups, product risks, test environments, invisible boundaries, transitions, conversions, time, and more.

2. Assuming limits and classes: It’s important not to make assumptions about the limits and classes that should be used in testing. The actual limits and classes may differ from what was originally assumed, and testing based on incorrect assumptions can result in missing critical bugs.

3. Ignoring user behavior: Testers must take into account the different ways in which users interact with the software. Factors such as user expectations, use cases, and workflows can all have an impact on the software’s behavior.

4. Over-reliance on BVA and ECP: While BVA and ECP are useful techniques, they shouldn’t be the only testing methods used. Other testing methods, such as exploratory testing and usability testing, can provide valuable insights that may be missed using BVA and ECP alone.

5. Not considering edge cases: Finally, it is important to test edge cases that may not be within the expected range of values. These edge cases may include unexpected inputs or user behavior and can help uncover critical bugs that may otherwise go unnoticed.

Boundary Value Analysis vs Equivalence Class Partitioning

Now that we have understood the various aspects of BVA and ECP; and how they can compliment each other. Let’s also throw some light on their differences:

BVA	ECP
Here, testing is always done on the boundaries of input values.	Here, the testing will be done according to the classes created. The input values can be taken from the boundary or from the middle region of the class.
It involves testing for valid inputs as well as invalid inputs.	The validity and invalidity of the input values depends on the partition. Either all values in the partition will be a valid input or an invalid input.
For eg. For testing boundaries for an age input form: 0 & 1 will be valid inputs and -1 will be an invalid input	

Automate your BVA and ECP tests, 10x faster

Conclusion:

In summary, BVA and ECP are not just testing techniques but are thought processes and approaches that can help testers to understand the system under test better and prioritize the tests based on the system risks and criticality, thus improving the overall test coverage, test plan, and strategy.