Software Quality Assurance and Testing Lecture - 04

ABDUS SATTER
LECTURER
INSTITUTE OF INFORMATION TECHNOLOGY
UNIVERSITY OF DHAKA

Dynamic Testing

BLACK BOX TESTING TECHNIQUES

Boundary Value Analysis (BVA)

- An effective test case design requires test cases to be designed such that they maximize the probability of finding errors. BVA technique addresses this issue. With the experience of testing team, it has been observed that test cases designed with boundary input values have a high chance to find errors. It means that most of the failures crop up due to boundary values.

- BVA is considered a technique that uncovers the bugs at the boundary of input values. Here, boundary means the maximum or minimum value taken by the input domain. For example, if A is an integer between 10 and 255, then boundary checking can be on 10(9,10,11) and on 255(256,255,254). Similarly, B is another integer variable between 10 and 100, then boundary checking can be on 10(9,10,11) and 100(99,100,101)

Boundary Value Checking (BVC)

- In this method, the test cases are designed by holding one variable at its extreme value and other variables at their nominal values in the input domain.
- The variable at its extreme value can be selected at:
  - Minimum value (Min)
  - Value just above the minimum value (Min^{+})
  - Maximum value (Max)
  - Value just below the maximum value (Max^{-})

Boundary Value Checking (BVC)

- Let us take the example of two variables, A and B. If we consider all the above combinations with nominal values, then following test can be designed:
  1. A_{nom}, B_{min}
  2. A_{nom}, B_{min+}
  3. A_{nom}, B_{max}
  4. A_{nom}, B_{max-}
  5. A_{min}, B_{nom}
  6. A_{min+}, B_{nom}
  7. A_{max}, B_{nom}
  8. A_{max-}, B_{nom}
  9. A_{nom}, B_{nom}

Robustness Testing Method

- The idea of BVC can be extended such that boundary values are exceeded as:
  - value just greater than the Maximum value (Max^{+})
  - value just less than Minimum value (Min^{-})
- When test cases are designed considering the above points in addition to BVC, it is called robustness testing.

Robustness Testing Method

1. A_{nom}, B_{min}
2. A_{nom}, B_{min+}
3. A_{nom}, B_{max}
4. A_{nom}, B_{max-}
5. A_{min}, B_{nom}
6. A_{min+}, B_{nom}
7. A_{max}, B_{nom}
8. A_{max-}, B_{nom}
9. A_{nom}, B_{nom}
10. **A_{max+}, B_{nom}**
11. **A_{min-}, B_{nom}**
12. **A_{nom}, B_{max+}**
13. **A_{nom}, B_{min-}**

Worst-Case Testing Method

- We can again extend the concept of BVC by assuming more than one variable on the boundary. It is called worst-case testing method.

1. A_{nom}, B_{min}
2. A_{nom}, B_{min+}
3. A_{nom}, B_{max}
4. A_{nom}, B_{max-}
5. A_{min}, B_{nom}
6. A_{min+}, B_{nom}
7. A_{max}, B_{nom}
8. A_{max-}, B_{nom}
9. A_{nom}, B_{nom}
10. A_{min}, B_{min}
11. A_{min+}, B_{min}
12. A_{min}, B_{min+}
13. A_{min+}, B_{min+}
14. A_{max}, B_{min}
15. A_{max-}, B_{min}
16. A_{max}, B_{min+}
17. A_{max-}, B_{min+}
18. A_{min}, B_{max}
19. A_{min+},B_{max}
20. A_{min}, B_{max-}
21. A_{min+}, B_{max-}
22. A_{max}, B_{max}
23. A_{max-}, B_{max}
24. A_{max}, B_{max-}
25. A_{max-}, B_{max-}

Example

- A program computes a^b where a lies in the range [1,10] and b within [1,5]. Design test cases for this program using BVC, robust testing, and worst-case testing methods.

	a	b
Min value	1	1
Min^+ value	2	2
Max value	10	5
Max^- value	9	4
Nominal value	5	3

BVC

Test Case ID	a	b	Expected Output
1	1	3	1
2	2	3	8
3	10	3	1000
4	9	3	729
5	5	1	5
6	5	2	25
7	5	4	625
8	5	5	3125
9	5	3	125

Robust Testing

	a	b
Min value	1	1
Min^{-} value	0	0
Min^{+} value	2	2
Max value	10	5
Max^{+} value	11	6
Max^{-} value	9	4
Nominal value	5	3

Robust Testing

Test Case ID	a	b	Expected output
1	0	3	Invalid input
2	1	3	1
3	2	3	8
4	10	3	1000
5	11	3	Invalid input
6	9	3	729
7	5	0	Invalid input
8	5	1	5
9	5	2	25
10	5	4	625
11	5	5	3125
12	5	6	Invalid input
13	5	3	125

Worst Testing

	a	b
Min value	1	1
Min^{+} value	2	2
Max value	10	5
Max^{-} value	9	4
Nominal value	5	3

Worst Testing

Test Case ID	a	b	Expected Output
1	1	1	1
2	1	2	1
3	1	3	3
4	1	4	1
5	1	5	1
6	2	1	2
7	2	2	4
8	2	3	8
9	2	4	16
10	2	5	32
11	5	1	5
12	5	2	25
13	5	3	125
14	5	4	625
15	5	5	3125
16	9	1	9
17	9	2	81
18	9	3	729
19	9	4	6561
20	9	5	59049
21	10	1	10
22	10	2	100
23	10	3	1000
24	10	4	10000
25	10	5	100000

Thank You

END OF CHAPTER