



**Software Quality Assurance
(4022)
Group Assignment
Report**

**Higher National Diploma in Information
Technology
ATI –Tangalle**

Group Members

Registration number :-

TAN/IT/2021/F/0019-S.V.P.R Sandamini

TAN/IT/2021/F/0058-M.K.D.Shrivanthika

TAN/IT/2021/F/0070-D.A.T.H.Kawshani

TAN/IT/2021/F/0105- K.A.N.K.Vijerathna

Contents

Group Members	2
Introduction.....	4
Black Box Testing	4
Definition	4
Methodologies.....	4
Advantages.....	5
Disadvantages	5
Applications	5
White Box Testing.....	6
Definition	6
Methodologies.....	6
Advantages.....	6
Disadvantages	7
Applications	7
Conclusion	7

Black Box and White Box Testing Techniques

Introduction

Software testing is a crucial process in the software development lifecycle. It ensures that the software meets its requirements and performs its intended functions. Two fundamental testing techniques are Black Box Testing and White Box Testing. This report provides a comprehensive overview of these techniques, their methodologies, advantages, disadvantages, and their applications in the software testing process.

Black Box Testing

Definition

Black Box Testing is a software testing method in which the internal structure, design, and implementation of the software being tested are not known to the tester. The focus is on testing the functionality of the software by providing inputs and examining the outputs without considering how the program processes the inputs.

Methodologies

- Equivalence Partitioning :- Divides input data into equivalent partitions that can be used to reduce the total number of test cases while maintaining coverage.
- Boundary Value Analysis :- Focuses on the values at the boundaries. It tests the edges of input ranges because errors often occur at the boundaries.
- Decision Table Testing :- Uses a table to represent combinations of inputs and their corresponding outputs, helping in identifying and testing all possible scenarios.
- State Transition Testing :- Examines the different states of a system and tests the transitions between these states based on inputs.
- Use Case Testing :- Derived from use case documents to ensure that all possible user interactions are tested.

Advantages

- No Knowledge of Internal Code Required :- Testers do not need to understand the code, making it accessible to non-technical testers.
- Unbiased Testing :- Focuses solely on requirements and functionality, reducing bias towards the implementation.
- User Perspective :- Tests the system from the end-user's perspective, ensuring that the software behaves as expected in real-world scenarios.

Disadvantages

- Limited Coverage :- Since it does not consider the internal code, some parts of the code may not be tested thoroughly.
- Inefficiency in Identifying Certain Defects :- May miss logical errors and hidden defects within the code that are not apparent through external testing.

Applications

- Acceptance Testing :- Verifies the software against user requirements.
- System Testing :- Ensures that the entire system functions as intended.
- Regression Testing :- Checks that new changes have not adversely affected existing functionality.

White Box Testing

Definition

White Box Testing, also known as Clear Box Testing, Open Box Testing, or Glass Box Testing, involves testing the internal structures or workings of an application. The tester requires knowledge of the internal code structure to design test cases that evaluate the functionality of specific paths, branches, and conditions within the code.

Methodologies

- Statement Coverage :- Ensures that every possible statement in the code has been executed at least once.
- Branch Coverage :- Tests all possible branches or decision points in the code to ensure that every path has been taken.
- Path Coverage :- Ensures that all potential paths through the code have been tested.
- Condition Coverage :- Tests all logical conditions, ensuring that each condition evaluates to true and false.
- Loop Testing :- Focuses on validating loops, including their initialization, execution, and termination.

Advantages

- Thorough Testing :- Provides comprehensive coverage of the internal logic, paths, and conditions.
- Optimization :- Helps in optimizing code by identifying redundant or unnecessary code.
- Error Identification :- More effective in finding hidden errors, logical errors, and security vulnerabilities.

Disadvantages

- Requires Detailed Knowledge of Code :- Testers need to have programming skills and understand the code, making it less accessible to non-technical testers.
- Time-Consuming :- Designing and executing tests for every possible path and condition can be very time-consuming.
- Less Focus on User Perspective :- May not adequately address how the software functions from the end-user's perspective.

Applications

- Unit Testing :- Testing individual units or components of the software.
- Integration Testing :- Ensuring that integrated modules work together as expected.
- Security Testing :- Identifying vulnerabilities and ensuring the software is secure from internal threats.

Conclusion

Both Black Box and White Box Testing are essential in ensuring the quality and reliability of software. Black Box Testing is ideal for validating the functionality and behavior of the software from an end-user perspective, while White Box Testing is crucial for ensuring the internal code quality and security. Employing both techniques in a complementary manner can provide a more comprehensive testing strategy, leading to robust and reliable software products.