Black Box and White Box Testing

Abstract

This report explains software testing by reviewing the methodologies and techniques used in Black Box and White Box testing. The differences of Black Box and White Box testing are compared, by analysing the techniques used and covering the advantages and disadvantages of each type. A conclusion is drawn based of this analysis.

Introduction to Software Testing

When a piece of software or application is created, it must first be tested to ensure it meets a client's requested specification or fulfil a need. Testing is considered part of the software development life cycle, as stated by Nidhra *et al* (2012)“Software testing is involved in each stage of software life cycle”to make sure there are no errors or bugs. Software testing is essentially just fault finding. During this stage the software can be reviewed and problems that need resolving can be flagged. This improves functionality and provides a more polished product. There are two main types of software testing that are practiced in the industry, Black Box and White Box. Each test has a different methodology and approach, which can be utilised to fulfil the appropriate specification needed.

What is Black-Box Testing?

Black Box Testing, or behavioural testing, is a software testing method that requires no knowledge of the internal code design or structure. Black Box testing essentially interacts with the user interfaces and analyses how they behave. This method is predominantly used to find interface or performance errors. Black Box testing can be functional or non-functional. Testers use techniques such as a boundary value test, which is undertaken to test the acceptable range of input values, with predictable outputs and helps reduce the total number of test cases.

The advantages of Black Box testing are that the software is tested from a user's point of view or independent source - this eliminates developer bias. Black Box testing is important for user feedback as they will be unbiased and will “help expose any ambiguities or inconsistencies in the requirements specifications” (Nidhra *et al*, 2012). Black Box testing is limited by the number inputs that can be tested therefore many branches of the software will not be tested - this is a major disadvantage to black box testing.

What is White-Box Testing?

White Box Testing or structural testing is a software testing method that requires the tester to have knowledge of the programming language the software has been created with. In addition to having the ability to understand the internal source code structure and its functionality. The tester will use information in the internal source code, to generate test cases by passing both legal and illegal inputs, which will ascertain the corresponding outputs are free of errors (Mariani *et al*, 2014). An advantage to White Box testing, is that it can be carried out at an early stage and is part of the software development life cycle, as mentioned by Lammermann “White-box testing is an important method for the early detection of errors during software development (Lammermann *et al*, 2005). White box testing is also very comprehensive and using this method can cover many, if not all branches of a piece of software when testing. An arguable disadvantage is the knowledge of the tester “the test is precise only if the tester recognizes what the program is supposed to do” (Nidhra *et al*, 2012) because white box testing is so comprehensive the tester’s knowledge and resources must also be comprehensive. This could also be seen as a distinct advantage in the eyes of someone knowledgeable enough to perform testing to high standard.

Comparison

Black Box Testing is a functional test and focuses on external behaviour. White Box is a structural test with the goal of testing specific paths of the code as explained by Nidhra “the goal of selecting such test cases is to cause the execution of specific spots in the software entity, such as specific statements, program branches or paths” (Nidhra *et al*, 2012). Black Box requires no knowledge of programming or access to the source code according to Henard “Black-box testing has the advantage of not requiring source code, thereby obviating the need for instrumentation and source code availability” (Hernard *et al*, 2016). The tester will only use requirement specification knowledge to verify inputs with the expected outputs. This is usually carried out by a dedicated tester and not the developer. On the other side, White Box testing is carried out by the developer and does require the source code, in addition the tester must have knowledge of the code and not just the requirement specifications as with Black Box.

Conclusion

Overall Black Box and White Box testing utilise different methodologies and approaches therefore, require access to different resources to be carried out effectively. Each method holds its own strength and is dependent on the type requirements of the test desired, in order to determine which method would be best suited to find the most errors or faults. Midian spoke about this in 2002 “the type of test depends on the requirements of the organization” (Midian, 2002). Although the study is from 2002 it still holds true. When developing software if the resources are accessible, both methods and their relative techniques should be carried out, to deliver a complete and functional piece of software. That is the interface and internal structure of the software perform together without bugs or errors.

Bibliography

Henard, C., Papadakis, M., Harman, M., Jia, Y. and Le Traon, Y. (2016) Comparing White-box and Black-box Test Prioritization. *2016 Ieee/acm 38th Ieee International Conference on Software Engineering* [online]., pp. 523-534. [Accessed 18 February 2019].

Lammermann, F. and Wappler, S. (2005) Benefits of Software Measures For Evolutionary White-box Testing. *Gecco '05 Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation* [online]., pp. 1083-1084. [Accessed 20 February 2019].

Mariani, L, Pezzè, M, Riganelli, O and Santoro, M (2014) Automatic testing of GUI‐based applications. *Software Testing, Verification and Reliability* [online]. 24 (5), pp. 29-50. [Accessed 18 February 2019].

Midian, P. (2002) Perspectives on Penetration Testing — Black Box Vs. White Box. Network Security [online]. 2002 (11), pp. 10-12. [Accessed 20 February 2019].

Nidhra, S. and Dondeti, J. (2012) BLACK BOX AND WHITE BOX TESTING TECHNIQUES –A LITERATURE REVIEW. *International Journal of Embedded Systems and Applications (Ijesa)* [online]. 2 (2), pp. 29-50. [Accessed 18 February 2019].