**DIFFERENCE BETWEEN BLACK BOX AND WHITE BOX TESTING.**

|  |  |
| --- | --- |
| **WHITE BOX TESTING** | **BLACK BOX TESTING** |
| The main objective is to test the internal structure of the application. | The main objective of this testing is to test the external Behavior of the application. |
| Tester should have the knowledge of internal structure and how it works. | This can be performed by a tester without any coding knowledge of the AUT (Application Under Test). |
| Testing can be done at an early stage before the GUI gets ready. | Testing can be performed only using the GUI. |
| This testing is more thorough as it can test each path. | This testing cannot cover all possible inputs. |
| Some testing techniques include Conditional Testing, Data Flow Testing, Loop Testing etc. | Some test techniques include Boundary Value Analysis, Equivalence Partitioning, Error Guessing etc. |
| Test cases should be written based on the Detailed Design Document. | Test cases should be written based on the Requirement Specification. |
| Test cases will be simple with the details of the technical concepts like statements, code coverage etc. | Test cases will have more details about input conditions, test steps, expected results and test data. |
| This is the responsibility of the Software Developers. | This is performed by professional Software Testers. |
| Programming and implementation knowledge is required. | Programming and implementation knowledge is not required. |
| Is mainly used in the lower levels of testing like Unit Testing and Integration Testing. | Mainly used in higher level testing like Acceptance Testing, System Testing etc. |
| This is more time consuming and exhaustive. | This is less time consuming and exhaustive. |
| It is easy to identify the test data as only a specific part of the functionality is focused at a time. | Test data will have wide possibilities so it will be tough to identify the correct data. |
| Main focus will be on how the application is built. | Main focus of the tester is on how the application is working. |
| Almost all the paths/application flow are covered as it is easy to test in parts. | Test coverage is less as it cannot create test data for all scenarios. |
| Code access is required. | Code access is not required. |
| Early defect detection is possible. | Defects are identified once the basic code is developed. |
| Helps to identify the hidden errors and helps in optimizing code. | Code related errors cannot be identified or technical errors cannot be identified. |