Week 3 Learning Activity - Testing Techniques

CST313: Software Testing

Professor:  Michael Hayden

Shaun Hoadley

21 June 2021

Throughout the software development life cycle, it is crucial to continually test the system to ensure each component functions properly both on their own and with each other to ensure they meet all of the requirements and the customer’s needs. As not all of the parts of a system may be fully developed, one needs to be able to simulate the way they behave. This simulation is accomplished by using a test bed. The test bed, according to Spillner, Linz, & Schaefer, uses stubs to emulate the inputs and outputs needed to analyze the test object (2014). The test bed may be used for any of the testing techniques, whether it be black box, white box, or experience (intuition)-based testing.

Black box testing is a testing technique used when the tester has no, or very limited, knowledge of the source code of the test object. Black box testing is also referred to as requirements-based, functional, behavioral, or specification-based testing. The purpose of black box testing is to observe the way the test object behaves from an external view (Point of Observation or PoO). Similarly, all influence associated with the test object is done with an external Point of Control (PoC). With black box testing, it is important to test valid and invalid inputs alike. The test cases for black box testing are derived from the specifications of the system and how it is intended to be used.

White box testing, also referred to as glass or open box testing, is used for testing the design of the test object. With white box testing, the tester has access to the source code and the PoO and PoC are contained within the test object to analyze the internal flow of the program. For this reason, the tester is able to inject conditions that should not be possible from the interfacing components in order to cause failure conditions. White box test cases are created with the assistance of the code structure itself. White box testing may also be called structural testing due to its consideration of the component hierarchy, control flow, and data flow.

Experience-based or intuitive based testing is usually a type of black box testing the test cases are not necessarily done systematically, but rather by the tester’s experience, skills, and knowledge in particular areas of programming. An example being someone who primarily works in programming network communication interfaces would be able to develop test cases involving network communication test objects more effectively than one who primarily deals with GUIs (Graphical User Interfaces).

Each of the testing techniques are used for different purposes and therefore not entirely effective by themselves for complete system testing. When used in conjunction with each other, they can significantly decrease the chances of post-implementation issues arising, thus reducing long-term costs for the development team and the client.

**References**

Spillner, A., Linz, T., & Schaefer, H. (2014). [*Software testing foundations: A study guide for the certified tester exam* (4th ed.)](https://ashford.instructure.com/courses/86077/modules/items/4350686). Rocky Nook.