**Week 3 Discussion 2: Testing Techniques Types, and Levels**

Shaun Hoadley

CST499: Capstone for Computer Software Technology

Professor Amr Elchouemi

January 27, 2022

**Testing Techniques Types, and Levels**

**When is the right time to stop testing?**

There is no easy answer to the question of when is the right time to stop testing. In a perfect world, testing should never end because there is no such thing as perfect software. However, in the real world, testing is stopped when it is "good enough," or as stated by Tsui et al., the software "meets the requirements and is fit to use" (2018). More often than not, testing is stopped due to resource budgeting running out. What I mean by that is when the cost of time, money, or personnel no longer outweighs the benefit of continued testing.

**Various testing techniques and levels**

There are four levels of software testing: Unit testing, Integration testing, System testing, and Acceptance testing. Unit testing consists of testing the individual components to ensure they fulfill the required functionality. Integration testing is used for testing the data flow between components to ensure they work together correctly. System testing evaluates the project's functional and non-functional needs in an isolated or stand-alone environment. Acceptance testing checks the requirements of a specification or contract are met, usually by the client, and is ready for deployment.

There are a variety of testing techniques available, but they can be grouped into three major categories: Blackbox testing, Whitebox testing, and Experience-based or Intuitive-based testing. Blackbox and Whitebox testing are both systematic testing methods, whereas Experience-based or Intuitive-based testingis non-systematic. Blackbox testing is a method of testing in which the internal structure of the program or code is not necessarily known and is generally the type of testing that is done by testers. Some Blackbox testing techniques include the following:

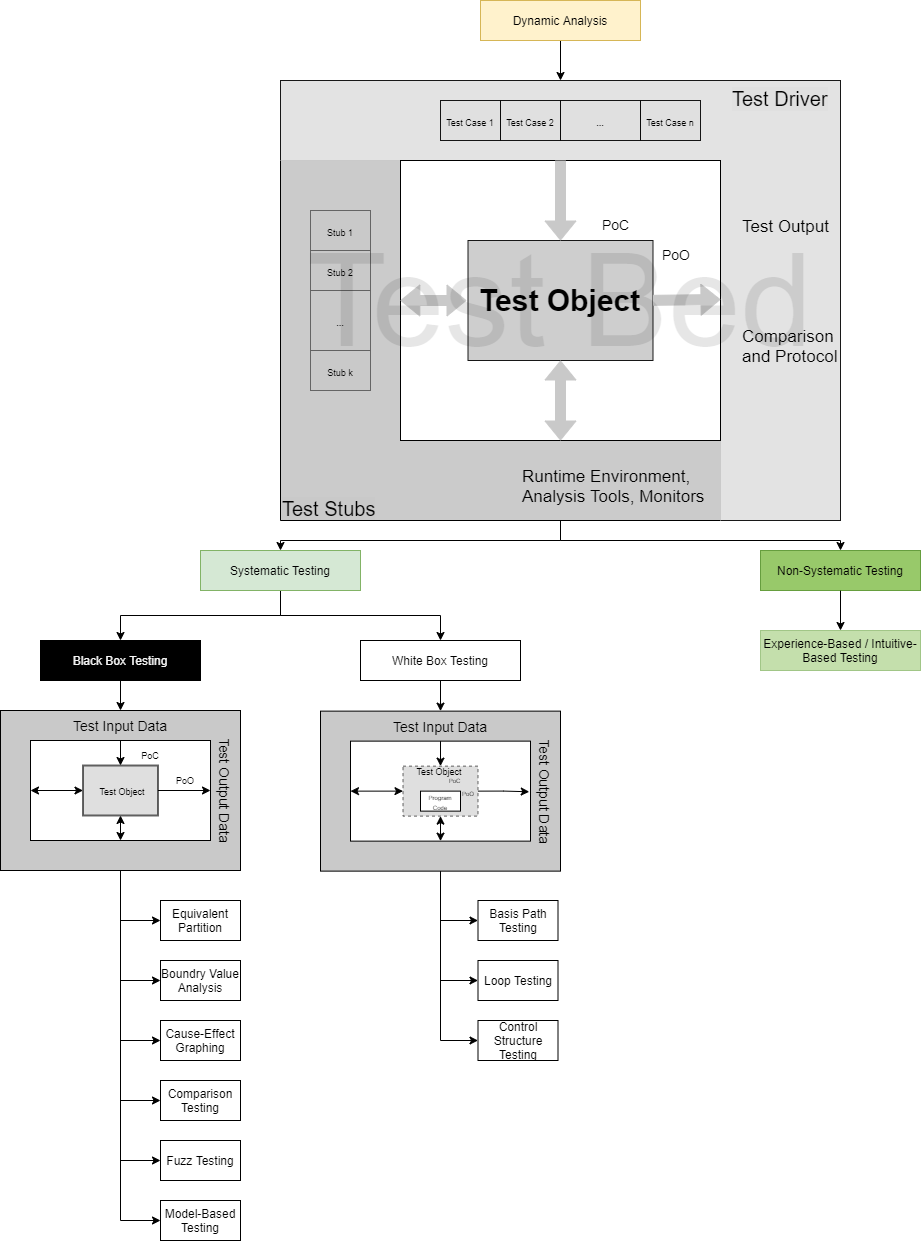
* Equivalent Partition
* Boundary Values Analysis
* Cause-Effect Graphing
* Comparison Testing
* Fuzz Testing
* Model-Based Testing

In contrast, Whitebox testing (sometimes referred to as Clearbox) is usually performed by developers rather than testers because it involves analysis of the internal structure. Some Whitebox testing techniques are as follows:

* Basis Path Testing
* Loop Testing
* Control Structure Testing

**How coverage data supports a determination of when to stop testing**

Coverage data can be used to help determine when to stop testing because it gives one a picture of what requirements, both functional and non-functional, have been tested and how much they have been tested. When you have one-hundred percent test coverage, then all requirements have been tested, and it is usually safe to end testing.

**Concept Map**

**References**

Tsui, F., Karam, O., & Bernal, B. (2018). Essentials of software engineering (4th ed.). Jones & Bartlett Learning.