Testing Study Points Descriptive Essay – Ben Jones

1. The 4 Agile testing quadrants

   When we talk about the 4 agile testing quadrants we can cover all aspects of testing an application that we should need to think about. They fall into two main categories. Business facing tests, and technology facing tests. As the names suggest, these two types of tests are relevant to the people that work closely with these sides of the application, whatever that might be.

   Q1 and Q4 are the technology facing quadrants, Q1 contains unit level tests, and supports developers. These unit tests can also be automated.Q4 contains the latter stages of the technology tests with system or opeational acceptance tests, including performance, load, stress, maintanability and scalability testing. These can also be automated with specialist tools.

   Q2 and Q3 are the business facing quadrants, Q2 are system level tests that follow how the software should behave, these include functional tests. In Q3 we have the system/user acceptance tests, which are based on scenarios and work flows. These are manual tests.

2. System Testing

   In system testing we are looking at testing the complete, integrated software system. This is to check that it complies with the functional requirement specifications (FRS) that have been provided and is the last step before acceptance testing. It is a black box method of testing, so it is not nessecary to have a vast knowledge of the inner workings of the software, just to be able to confirm that it does what it should do. Some of the main aspects covered are, Functionality, Inter-Operability, Performance, Scalability, Stress, Load and Stability, Reliability, Regression and Regulatory compliance. In two phases, system test plan and system test cases, we could have four stages, Prepare, Review, Rework and Baseline. We would the carry out the system test. These tests are normally performed by an independent team to keep their work unbiased.

3. Exploratory Testing

   Exploratory testing is the practice of learning about, designing tests for, and executing those tests, on a system. It is a black box testing method where the tester will learn things about a system , and use his ir her knowledge and experience to find different things to test the application for. This method of testing relies solely on the thinking and commitment of the tester to find ways to make good tests of the code and perform them. The testers must be skilled at finding cases to test and then looking for any defects there may be. It's almost the opposite of the scripted testing that we normally use. It's freestyle!

   It has benefits and drawbacks, like everything. It needs less preparation and is quick at finding results, and also keeps testers very engaged, but also tends to be less documented and harder to review/repeat, because of it's on the fly nature.

   It is useful where requirements are incomplete, because faults are found quickly.