Lukasz Koziarski

System Testing - Study Point Exercise
Descriptive Essay

The Agile Testing Quadrants are an Agile Testing matrix, used to help teams perform Agile testing in an agile context better. The 4 quadrants are technology facing tests that support the team (Q1), business facing tests that support the team (Q2), business facing tests that critique the product (Q3) and technology facing tests that critique the product (Q4). The numbering of the quadrants does not imply that there is an order in which to use the quadrants. The first and second quadrant focus more on code that has not been written yet, while quadrants 3 and 4 focus on code already having been written.

The first quadrant, technology facing tests that support the team, focuses on developing quality code and automated testing. This quadrants practices can act as a guideline for developers that are about to develop code. Quality code can be achieved by creating Unit Tests before writing code this will make code more bug free as well as give the developer a better idea of what is to be developed.

The second quadrant, business facing tests that support the team, is meant to get the requirements. This is done by creating automated and manual tests. This quadrant focuses more on the developer, the tester and the customer co-ordinating. Figuring out workflows and creating test cases that apply to the business logic.

The third quadrant, business facing tests that critique the product, is the creation of manual tests. These tests are an evaluation of the product and can be done with the customer as well. Doing tests such as exploratory testing, user acceptance testing and usability testing.

The fourth quadrant, technology facing tests that critique the product, test non-functional aspects of the product. This quadrant is conducted when the product is in a near complete state and no more functional code is to be written. Performance tests, infrastructure tests, scalability, load testing as well as security tests would be performed in this quadrant..

Exploratory Testing is the act of creating tests for software while performing tests. Meaning that you would start off with a few test cases and while figuring out what is happening you are coming up with new test cases. This focuses on less documentation and is also considered as a form of black box testing.

System Testing is when software is tested on a whole system. The system is tested against expected outcomes. The expected outcomes are determined from functional requirement specifications. This test method is considered a black-box test. There is a large array of tests that can be used in system tests, here are some examples: Usability testing, UI testing, security testing, scalability testing, recovery testing and many more.