

## **One page essay explaining subjects**

### **4 Quadrants of agile testing**

In his original blog about the Agile testing matrix<sup>1</sup>, Brian Marick describes what is now better known as the Quadrants of agile testing.

The 4 quadrants are enumerated, NOT because it implies any kind of order, but because it made referencing them in the Agile Testing<sup>2</sup> book easier.

It is imperative to understand, in the agile spirit, that the order is indeed unimportant.

The four quadrants is a kind of taxonomy to help teams plan their tests, and to make sure that they have the resources necessary for them.

The quadrants split up vertically and horizontally:

Vertically it splits up whether the tests are business or technology driven.

Horizontally it splits up whether the tests are supporting the dev. Team or critiquing the product.

### **System testing**

System testing is Evaluations made on a complete system, to verify that it meets the requirements set for the system.

Be it a software/hardware or integrated system, system testing evaluates not only the functionality of the system, but also the design and performance and many other aspects.

This means that remaining parts of this entire hand-in will exist underneath the umbrella we call "System Testing", as Exploratory Testing and Performance testing (Jmeter) is performed in system testing.

### **Exploratory Testing**

Exploratory testing is defined as simultaneous learning, testdesign and execution.

To decipher this a little bit: Exploratory testing doesn't rely on a designed test script, as with normal manual testing. Rather, it relies on the creativity and experience of the tester.

The result is often a very agile and fast test development, with less redundancy in testing. In comparison to scripted tests, exploratory testing is often referred to as "freestyle" testing, because the tester invents the tests as he is actually executing them.

A great advantage and disadvantage of exploratory testing, is the uncertainty that the performed test will test the same things during a second test. This can be good or bad depending on the project.

---

<sup>1</sup> <http://www.exampler.com/old-blog/2003/08/22/#agile-testing-project-2>

<sup>2</sup> <http://agiletester.ca/>