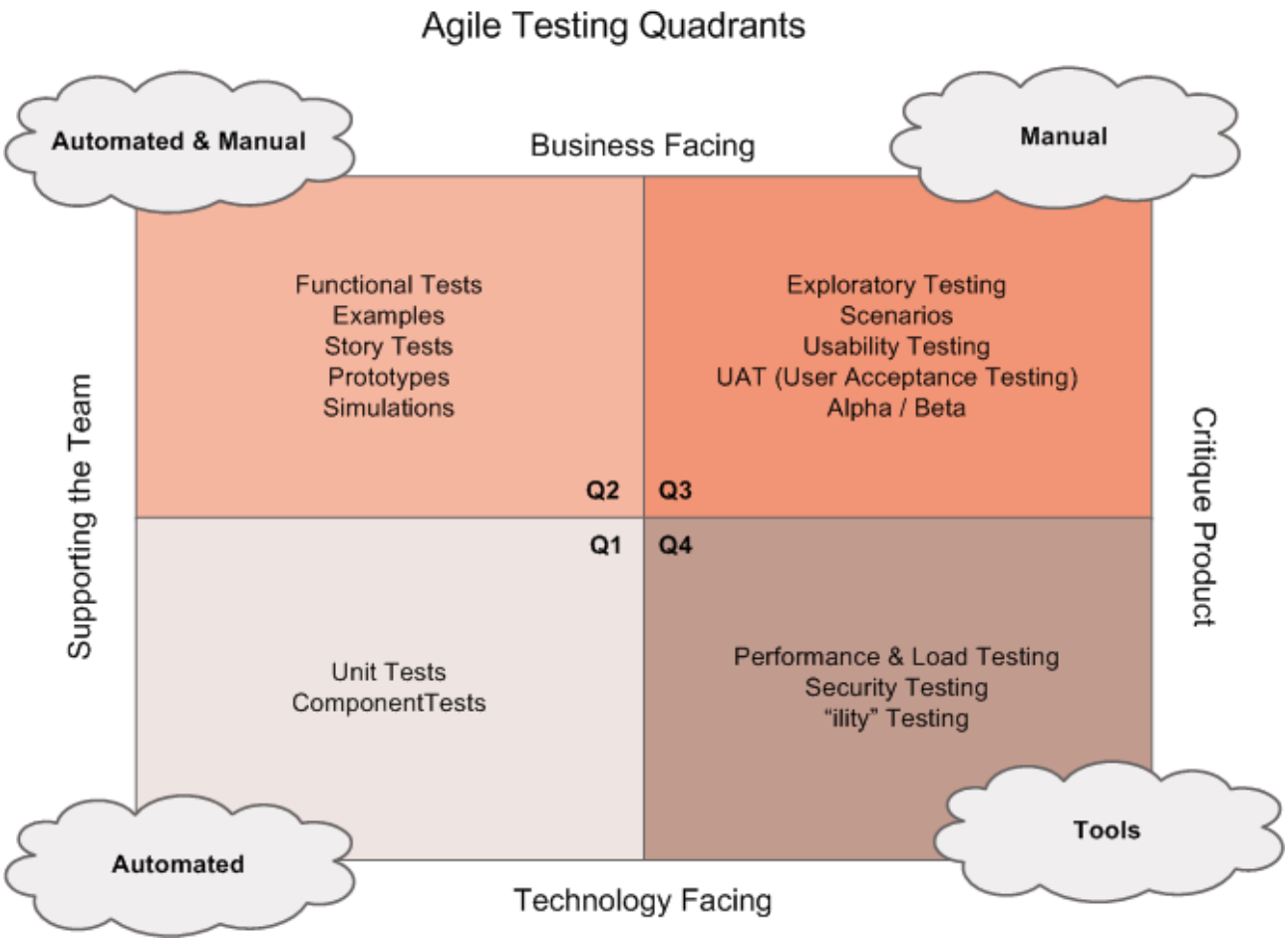
The 4 agile Testing Quadrants

The quadrant system does not imply any order. This means that you don't work through the quadrants from 1 – 4.



Figur 1 Agile Testing Quadrants

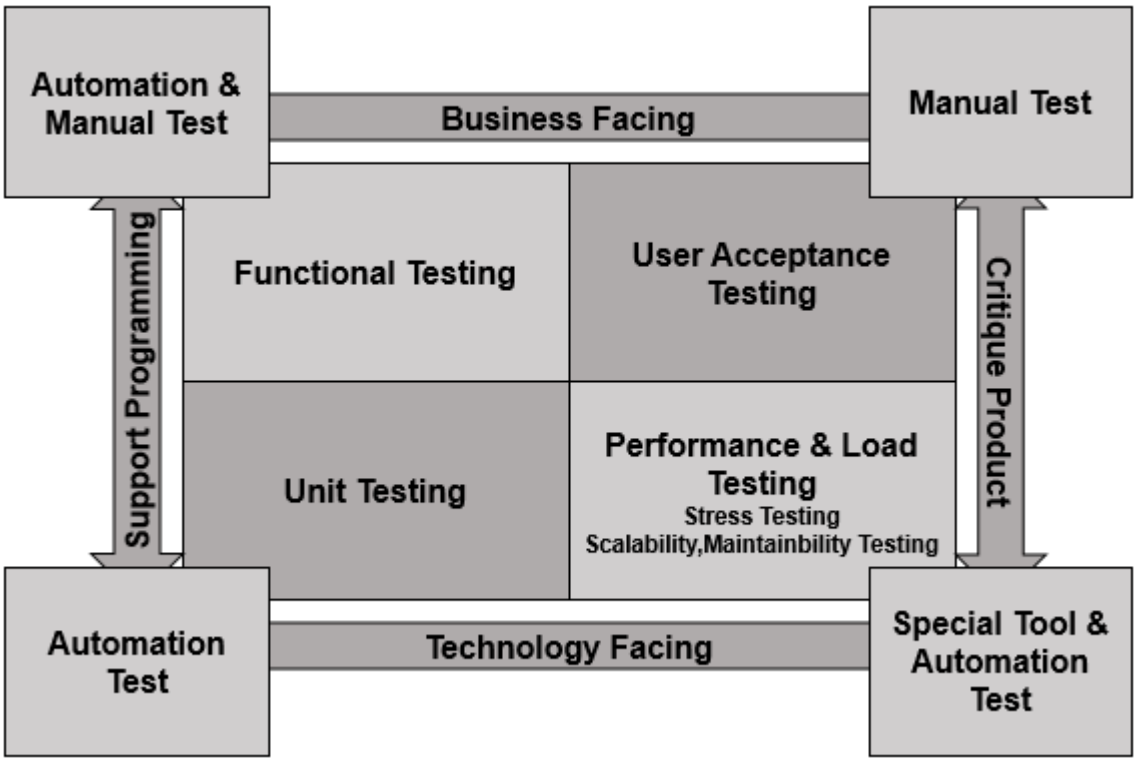
Most projects would start with **Q2** because here you would test the specifications of the product. The test here mainly system level testing meaning that there are either manual or automated, also known as functional testing.
Some examples would be story tests, simulations based test, user experience prototypes, of course as mentioned earlier they can be either manual or automated test. The purpose of the tests here is to create value for the customer.

Q1 is unit level testing in the programming world there are called unit tests. Here the tests consist of testing the components of the software, independently, and later combining the components together and then test them. The purpose here making sure that we are creating quality software.

Q3 at system or user acceptance level, business facing, contains tests exploratory testing, scenarios, process flows, usability testing, user acceptance testing, alpha testing, and beta testing. These tests are often manual and are user-oriented.

Q4 At system or operational acceptance level, technology facing performance, load, stress, and scalability tests, security tests, maintainability, memory management, compatibility and interoperability, data migration, infrastructure, and recovery testing. These tests are often automated.

We from the above, visualize as follows:



Figur 2 Visualization

https://www.tutorialspoint.com/agile_testing/agile_testing_quadrants.htm

http://istqbexamcertification.com/what-are-test-pyramid-and-testing-quadrants-in-agile-testing-methodology/

https://www.tutorialspoint.com/agile_testing/agile_testing_quadrants.htm

http://lisacrispin.com/2011/11/08/using-the-agile-testing-quadrants/

Made By Mazlum Dogan Sert & Ismail Cam

System Testing

Definition from: http://istqbexamcertification.com/what-is-system-testing/

In **system testing** the behavior of whole system/product is tested as defined by the scope of the development project or product.

- It may include **tests based on risks** and/or **requirement specifications**, business process, use cases, or other high level descriptions of system behavior, interactions with the operating systems, and system resources.
- System testing is most often the final test to verify that the system to be delivered meets the specification and its purpose.
- System testing is carried out by specialist's testers or **independent testers**.
- System testing should investigate both **functional** and **non-functional requirements** of the testing.

One most know that often done using black box methods, that means we don't see the code. We perform system testing before acceptance testing and after integrations testing. What a tester should ask himself doing system testing is "*does the system meets its original purpose"*, it would the same as "*does the system fulfil its requirements"*.

Exploratory Testing

As its name implies, exploratory testing is about exploring, finding out about the software, what it does, what it doesn't do, what works and what doesn't work.

The planning involves the creation of a test charter, a short declaration of the scope of a short (1 to 2 hour) time-boxed test effort, the objectives and possible approaches to be used.

Exploratory testing often performed as a black box testing technique, the tester learns things that together with experience and creativity generate new good tests to run.

https://www.tutorialspoint.com/software_testing_dictionary/exploratory_testing.htm

http://istqbexamcertification.com/what-is-exploratory-testing-in-software-testing/