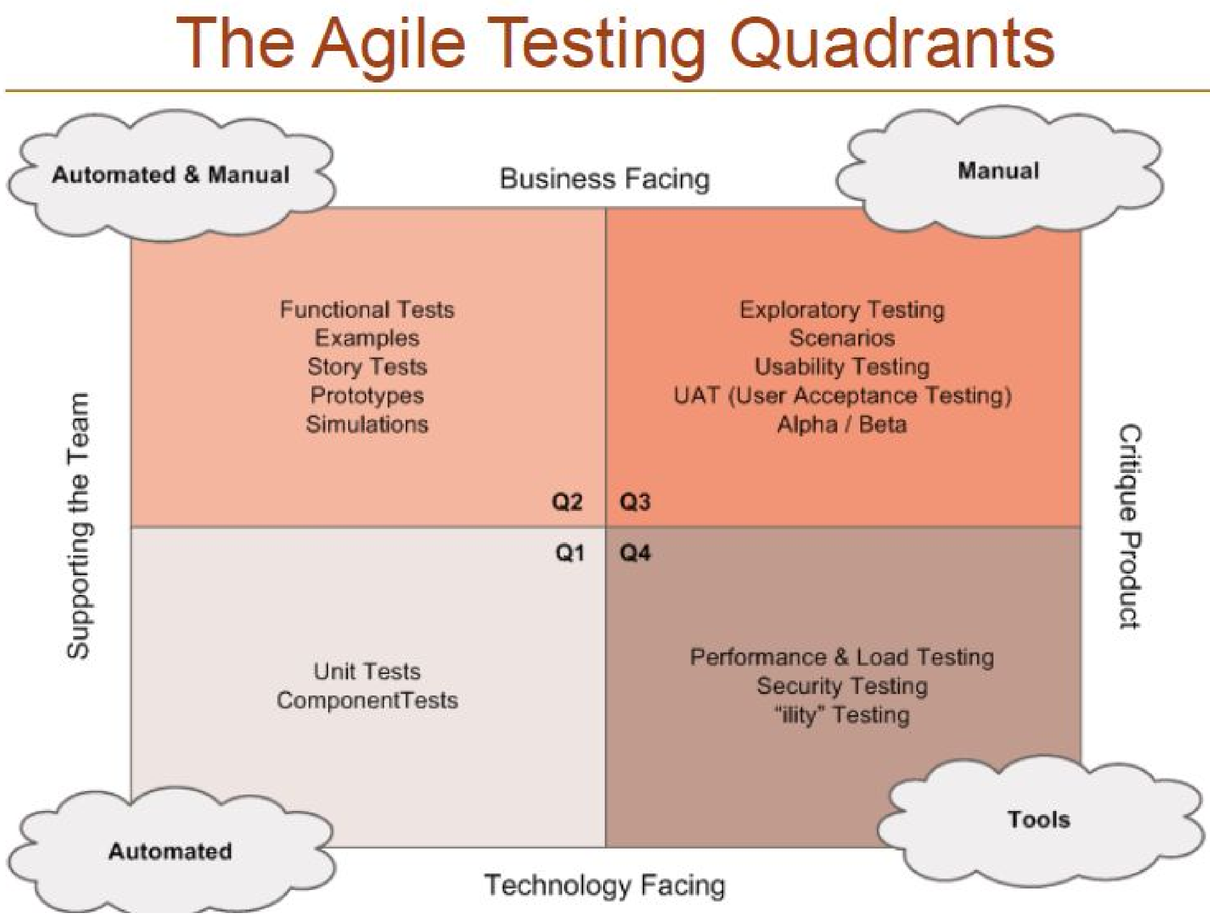
Exploratory testing - the best example I come across for this type of testing is solving a jigsaw puzzle. We keep trying pieces, some of us can even have a strategy finding the puzzle pieces for the angles or another type of approach. We keep trying(exploring) and simultaneously learn for our next move - *the puzzle changes the puzzling*. Exploratory testing involves of course an interacting with the product. According to the document we had to read the basic elements of ET are time, tester, product, mission and reporting. We have to keep trying and keep aligning ourselves to the mission so we would come with questions and design different tests and running the tests to get the answers so we repeat this process until we are done. Inside the head of every ET tester there different things for example:
1) how to design the next test - every test systematically explores the product
2) careful observation - much more enphasize on observation than compared to scripted testers
3) critical thinking - putting critical thinking and logic in the creation of tests
4) Diverse Ideas - new and different ideas needed to accomplish successful exploratory testing.
5) Rich resources - sooner or later ET testers build diverse inventory of tools for their testing adventures.
ET fits in situations that we are not sure what is the next test or we have the natural inclination.

System testing - is performed on the entire system based on the Functional and System Requirements basically taking into consideration Business/Functional and End-user requirements. It can be said that it is black-box testing. Considered the final testing in order to make sure the system that should be delivered is working properly. Testing the complete and fully integrated software product. Usually done only after System Integration test. It is important to point out that system testing is done in environment that resembles the production environment where the application will be deployed. System testing is performed on a server similar to the production server. As every other test this type of testing has a Plan. For example this could be:
What are the goals and objectives? Scope? Entry and Exit Criteria? Roles and Responsibility? and so on.
Any kind of other test types can be included in the system testing. For example:

- Exploratory testing
- Security testing
- Scalability testing and so on.

The 4 agile testing quadrants -
Quadrant 1 - technology - facing tests that support the team.
Quadrant 2 - business - facing tests that support the team.
Quadrant 3 - business - facing tests that critique the product.
Quadrant 4 - technology - facing tests that critique the product



Q1 - for example typical representation would be unit tests. We do them so they provide safety for the different units written , through them we improve design and maintainability without changing functionality. Quality is our goal here. We also can perform component testing so we make sure components work together properly.
Q2 - we encounter here the business-facing tests, we help the customer to clarify what actually they want. We got feedback if what we did is consider done. Tools for this quadrant could be brainstorming, words, ideas, tasks, flow diagrams. Most projects start with this quadrant because this is where you get the examples that turn into specifications
Q3 - recreate actual user experiences, learn as you test. Pair exploratory testing with customer
Q4 - tests Performance(How fast?) Stability(How long?) Reliability(How often?)
Q3 and Q4 require code to be created and deployed but most teams probably iterate through the quadrants. The quadrants are suppose to help teams plan their tests, their no strict rules what goes in which quadrant.