Fundamental test process are test planning and control, test analysis and design, test implementation and execution, evaluating and exit criteria and reporting. And finally, test closure. You go through these phases sequentially and iteratively, so it fits the agile mantra of Iterative, incremental and evolutionary development. As you break up the product development and test each block according to the fundamental test process.

As you develop and test in blocks it makes it easier to choose what to test. According to the 7 test principles exhaustive testing is not possible, therefore you must be aware and don’t get stuck testing everything. It’s a lot harder to over-test in small blocks. Some drawbacks may be some unforeseen bugs after stitching each block together, it may come from communication failure.

But as the principle defect cluster tells us is that 80% of defects are found in 20% of the modules. It may make it easier to identify them in agile development. And in each sprint, you may need to design new method of testing. The principle pesticide paradox tells us that test cases need to be regularly reviewed & revised, adding new & different test cases to help find more defects. It fits to agile development sprints and iterative nature, while you code you can also think, design and implement new testing for that specific module.

But as testing shows the presence of defects not absence of defects. So, after you design and test modules in sprints, defects may appear late in the development when you do a system test. Or it may show that you have a bug free program (no such thing) but you haven’t met the customer´s needs or requirements. That is a problem every development method must account for. According to the 7 testing principles, early testing may be a good tool to solve that problem. It pairs really well with agile development. As you are in constant communication across development team, business team, management and the costumer. The agile development principle to welcome change in requirements will is also a good way to deal early testing. By communicating with the customer and delivering working software frequently makes it easier to test the software in the context it will be used. As testing is context dependent.

V-model is another development method. It is popular among test managers as it very beneficiary for them. But it is rigid not flexible and agile is the opposite. It may cause some tension as their methodologies dos not complement each other.

Exploratory Testing is defined as simultaneous learning, test design and test execution.

If you continuously and frequently develop software part of a bigger system. Then exploratory testing may help you with the rapid development. It also will help you with automated testing. It focuses on new features while automated testing accounts for regression issues. The testing style of exploratory testing are unstructured, and it can focus on high risk areas and find possible issues with the product.

Test-driven development is a software development process you have short development cyle where requirements are specific test cases. It can cause extra work as in agile development may change requirements late in the development phase. But It may help the costumer communicate his or hers wishes more clearly.