

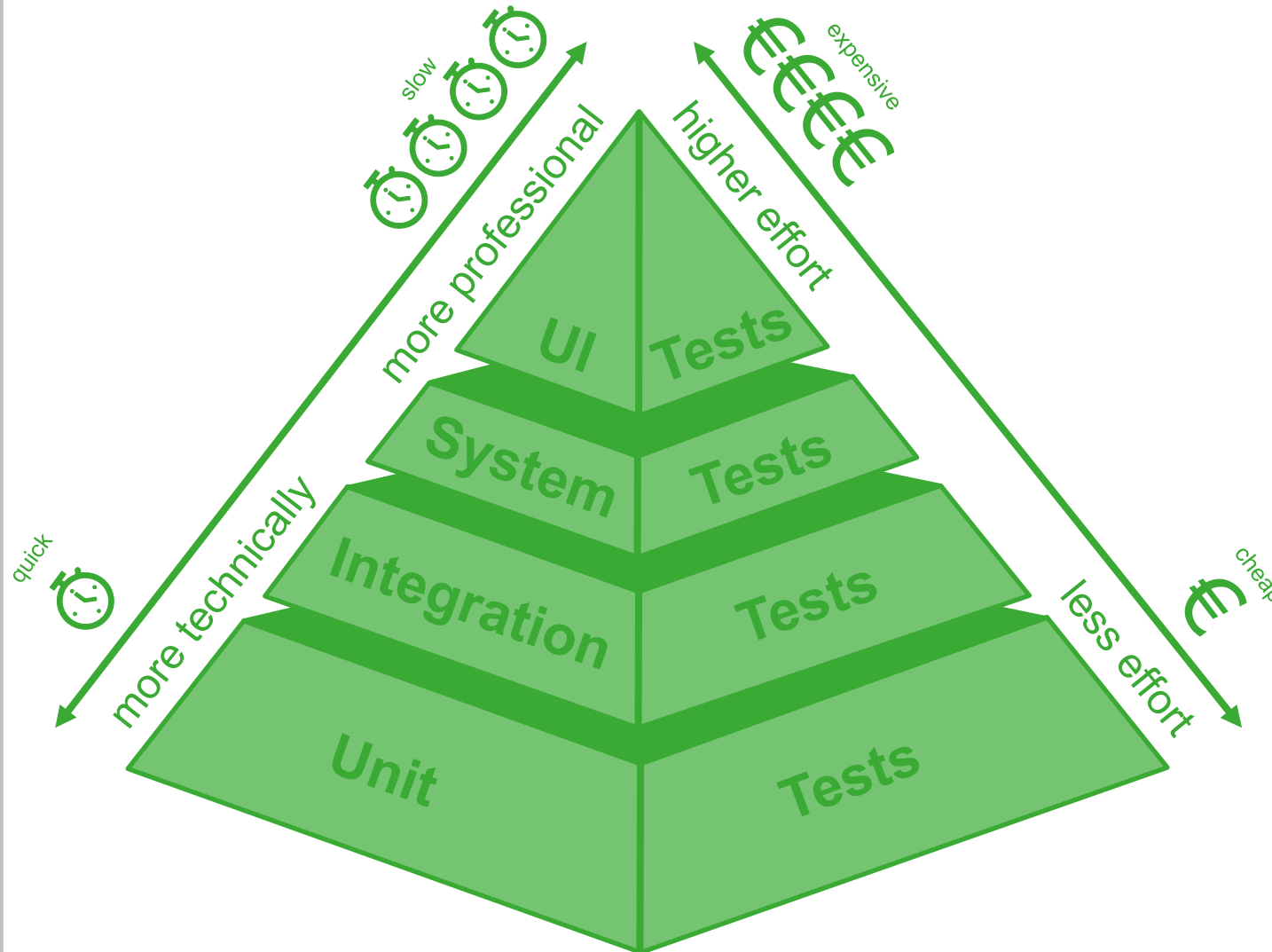
## The idea behind this:

A very important measure of progress in **agile** product development is **working software**, which means that the software has to be tested completely in short intervals. The test pyramid gives an indication of the extent to which tests should be performed in order to achieve fast feedback loops with short test run times and high robustness.

Granular, stable tests take up the largest part. Cross-cutting tests, which need to be adjusted frequently, occur rarely.

This means that the creation and maintenance effort is higher for the tests mentioned above and lower for those mentioned below. The further down tests are located, the more advisable automation is and vice versa.

Exclusively manual tests only make sense at the top or in exceptional cases.



## The four levels:

**UI Tests:** (User Interface Tests) End-to-End tests on the graphical user interface. Often explorative tests (can not be automated).

**System Tests:** inspection of the complete system against functional requirements (e.g. Use Cases) and non-functional requirements (e.g. security, reliability and performance).

**Integration Tests:** medium-scaled tests of the interaction of interdependent components.

**Unit Tests:** check the behavior of single components (technically executable, correct professional results).