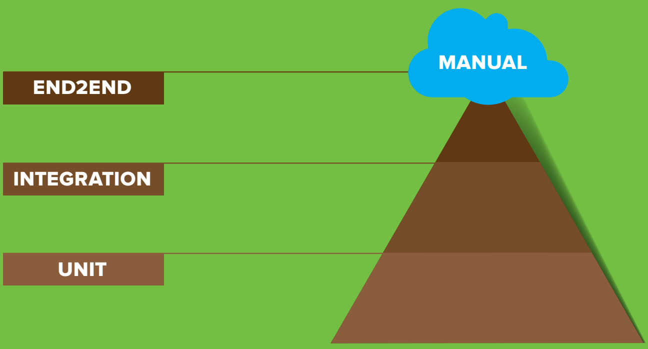
WE ♥ TEST AUTOMATION

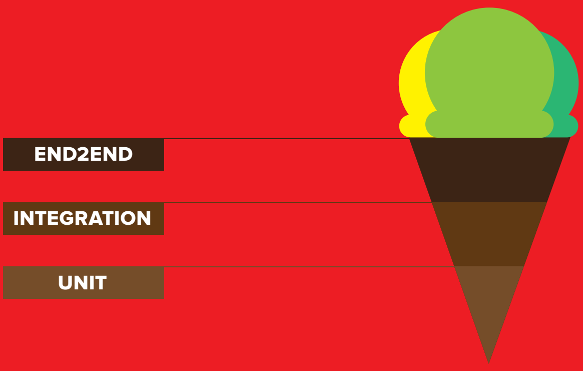
MIND THE AUTOMATION ANTI-PATTERNS!



TESTING PYRAMID

The goal of test automation is to accelerate the feedback cycle. In order to achieve this goal it is important to use the right mix of different test types. The testing pyramid is a concept that visualizes this distribution of testing types. The idea is that on the bottom of the pyramid there is a large base of fast, technology facing tests. Since these tests run fast, and are inexpensive to create, it's no problem to aim for a high coverage in this area. Higher up the pyramid there is a smaller set of business facing tests. These business facing tests are far more costly to develop, execute and maintain, so aim to keep this set as small as possible.

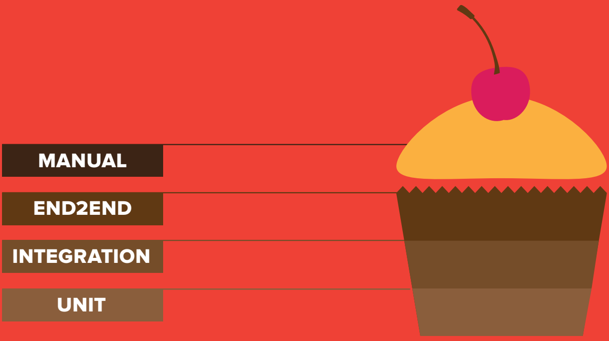
In practice, it's easy however to deviate from the distribution as depicted above. Below we describe some of the most common anti-patterns. Do you recognize one of these anti-patterns in your own automation setup? *Take action and call us!*



ICE-CREAM CONE

One of the most common anti-patterns is the so called ice-cream cone. What happens is that the test automation effort is focused on automating manual test scenario's. In nearly all cases this means that the automation is done through the GUI, over the entire application landscape. Tests on more isolated components are rare or non-existent.

The drawback of this approach is that you cannot achieve true fast feedback (there is still a considerable gap between code commits and the test feedback), the tests take long to develop and are difficult to maintain. Because all the tests require a fully integrated environment to run, test data management becomes increasingly challenging as well.



MANUAL

END2END

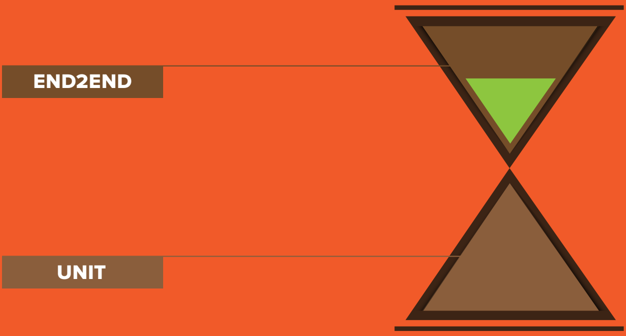
INTEGRATION

UNIT

CUPCAKE PATTERN

Testing isn't a team effort, everybody is involved but nobody is communicating. Developers are responsible for Unit and API tests. End2end tests are build by a test automation team. And exploratory tests are done by manual testers.

The teams do not agree on the scenarios that have to be tested or the level of test automation. This results in duplication - the same scenario ends up being automated at many different levels.

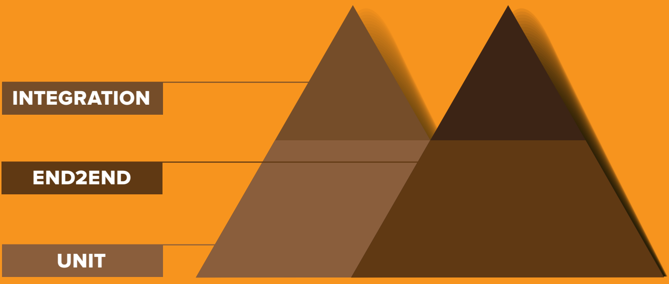


END2END

UNIT

HOURGLASS

By only using the familiar and easy test mythologies, you will end up with an hourglass model where a few testtypes are overrepresented. For instance, unit tests are made by developers, end2end tests are made by testers, and other testtypes are ignored altogether. In practice this means that either too much is being tested at the end to end level, or essential coverage is missing. Usually, it's a combination of both.



INTEGRATION

END2END

UNIT

DUAL PYRAMID

Focus on the testing pyramid is going as it is supposed to be. However, the different expertises aren't collaborating. Test professionals are operating in a separate team, which leads to developers and business who won't feel any responsibility for failures. This scenario will lead to a long feedback cycle because of the walls between the different expertises.

powered by Xebia

xebia.com