Ideal vs. Non-Ideal Testing Pyramids

Laurie Mailloux

WEB-430

Assignment 3.3

Ideal vs. Non-Ideal Testing Pyramids

It’s said that “Test Pyramid” is considered a metaphor that tells us how to group software tests into buckets of different granularity. While also giving an idea of how many test we should have in each one of these groups. The concept of the Test Pyramid has be around for a while, but teams still struggle to put it into practice properly.

Ideal Testing is a test in development that is used to catch bugs and errors in the early stages of automated testing. Ideal testing is broken up into a pyramid and that it is split into 2 different stages.

The first stage is Automated Unit Tests. This is on the base of the pyramid. When testing in this level, it allows us to check our work, while getting immediate feedback and letting the developers know exactly where all the bugs are. This concept allows us to test only a small amount of code which is independent code, and it doesn’t depend on anything else. If the test doesn’t work, you will know where to problem is located and the larger system will not be affected by the test. (Alger 2018) The biggest chunk is that 70% of tests written should be unit tests, which are primarily written by developers to test their own code and catch bugs early before reaching QA teams. These tests are crucial as they provide a fundamental of test coverage that reduces the number of major bugs found later in the test cycle.

Automated Service Test is the middle level that includes testing the Integration. Unit test aren’t enough to ensure quality of codebase, so integration testing is done. These tests should not appear as often as unit test. Integration test has been defined as test that validate interactions with external components. External components can be anything that lives outside of the application, but the app still depends on. The integration with external services should be tested also.

End-to-end testing is at the top of the automation pyramid. We want to ensure that our entire application is functioning as expected and that is where end-to-end testing will help with. End-to-end testing will test that your application is working right start to finish. The integration of the front end with the back end is tested. This can sometimes be time consuming and tedious, so Selenium, a testing framework that will automate interactions with browsers, can be used. It provides a domain-specific language that will allow developers to write test that will interact with a web applications running inside the browser. End-to-end test are at the top because they may be slow and fragile.

Here is a breakdown of what we have learned so far.

* The test automation pyramid is a framework that defines various types of tests and the number of times they should appear.
* Unit tests form the base of the test pyramid. They should be frequent, and they should run fast.
* Integration tests are the middle tier of the pyramid. These tests focus on interactions of your code with the outside world, such as databases and external services.
* End-to-end tests top the test pyramid. They’re written from the perspective of a user and should test that your entire application is functioning from front end to back end.

Non-deal testing is a pyramid but is show upside down. One person referred to it as “the ice-cream cone ati-pattern”. Its said to be opposite of what the Testing Pyramid is suppose to achieve. You may not achieve much with this approach; it’s considered a bad strategy. (But 2018) Having a large portion of manual testing compared to that of Unit Test cries for disaster. Development times increases because it takes much more time to do things manually.

Ideal-Testing defiantly sounds like the way to go, as it will allow you to develop a high-quality application. It also allows more testing on the base or front end with smaller amounts of cod and then you can build up into the integration and UI layers.

References

Alger, Leah. July 4, 2018. Adopting the ‘test pyramid model’ approach. Retrieved on July 26, 2020 from <https://www.softwaretestingnews.co.uk/adopting-the-test-pyramid-model-approach/>

But, Colin. Aug 4, 2018. Define Testing Strategy using the Testing Pyramid. Retrieved on July 26, 2020 from <https://medium.com/@Colin_But/define-testing-strategy-using-the-testing-pyramid-1dabee37e823>

Testim. November 25, 2019. Test Automation Pyramid: A simple strategy for your Test. Retrieved on July 26, 2020 from <https://www.testim.io/blog/test-automation-pyramid-a-simple-strategy-for-your-tests/>

Vocke. Ham. February 26, 2018. The Practical Test Pyramid. Retrieved on July 26, 2020 from <https://martinfowler.com/articles/practical-test-pyramid.html>