My Problem With the Pyramid



http://bit.ly/2FZ8y45

James Bach

james@satisfice.com

Big thank you to **Matthew Heusser**, whose irritating and helpful style of debate clarified my thinking on this subject, and inspired this talk.

Thank You Reviewers

- Damian Synadinos
- Laura Oniga
- Mark Raphael
- Paul Holland
- Michael Bolton
- Aaron Hodder
- Aleks Simic

"Test Automation Pyramid" (from Mike Cohn)



"Even before the ascendancy of agile methodologies like Scrum, we knew we should automate our tests. But we didn't. Automated tests were considered expensive to write and were often written months, or in some cases years, after a feature had been programmed. One reason teams found it difficult to write tests sooner was because they were automating at the wrong level. An effective test automation strategy calls for automating tests at three different levels, as shown in the figure below, which depicts the test automation pyramid."

https://www.mountaingoatsoftware.com/blog/the-forgotten-layer-of-the-test-automation-pyramid

Alister Scott's Model



https://watirmelon.blog/tag/testing-pyramid/

Alister Scott's Ice Cream Cone



Vocabulary Note

Test: An activity wherein

 a person
 evaluates a product in some way
 by learning about it through
 exploration or experimentation

(Not to be confused with a *check*, which involves only an algorithmic process of operating a product, collecting data, and applying pass/fail decision rules. Checks can be fully automated; tests can only be supported with automation. Tests may incorporate checks.)

• **High level test:** A test conducted on a complex assemblage of parts operating as an integrated whole

(As opposed to a "low-level" test conducted on some part of that assemblage, especially prior to integration. Unit testing is often done by automating and monitoring unit level checks.)

High Level Tests Have Special **Power**

- Bugs found with high level tests are more likely to be visible to the user and to impact user experience.
- High level testing reveals more about interactions and emergent properties of integrated subsystems.
- More and bigger dependencies means more sensitivity to buggy changes.
- Naturalistic sequences, timing, and data are easier to arrange with high level testing.

High Level Tests Give Special Headaches

- Bugs found with high level tests are more likely to be difficult to reproduce or trace back to code.
- High level testing is terrible for finding certain important problems that are easy to find with code review or unit-level checking.
- More and bigger dependencies means more sensitivity to *correct* changes, leading to false fails.
- Naturalistic sequences, timing, and data can still be expensive to create and maintain.

Inkscape Source Hierarchy



A product is like a planet.



Volume is an analogy for relative potential product risk. The model reminds us that complexity, interactions, volatility, visibility of trouble tends to grow toward the user.

by James Bach



Unlike the "ice cream cone" model, this is about product risks that deserve attention, *not counting freaking test cases*.

The top is the "living surface" of the product.

The top matters **intrinsically**.

The bottom makes the top **possible**.

This model sets up a tradeoff relationship instead of simply telling you what to do: "You will be judged by the surface, but you will succeed or fail by the depths"

by James Bach



Notice that the bottom-most gray part lies beneath units, yet developers rarely test anything on that level.

Which is probably okay, no? Or do you think you need to test Chrome and Linux, too?

by James Bach



Unlike any pyramid I've seen, this model reminds us that data is key, including subterranean data (i.e. production data on servers, built-in data, etc.).

Think of data as wind, water, and magma.

by James Bach



"up" at the user through the filter of many details, while having trouble looking down past the screen of whatever platform the product depends upon.

by James Bach



I created the "Round Earth" model as an antidote to once helpful ideas, such as the original pyramid, that were elevated into best practice dogma. It is an alternative that addresses similar concerns as did the original pyramid, **but helps you explain your reasoning better.**

I'm not saying you should "follow" my new model.

Heuristics are not to be followed.

Heuristics are to be applied as needed by skilled practitioners.