

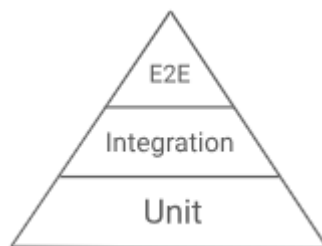
# SMURF: Beyond the Test Pyramid

Tuesday, October 15, 2024

*This article was adapted from a Google [Testing on the Toilet](#) (TotT) episode. You can download a [printer-friendly version](#) of this TotT episode and post it in your office.*

By Adam Bender

The [test pyramid](#) is the canonical heuristic for guiding test suite evolution. It conveys a simple message - *prefer more unit tests than integration tests, and prefer more integration tests than end-to-end tests.*

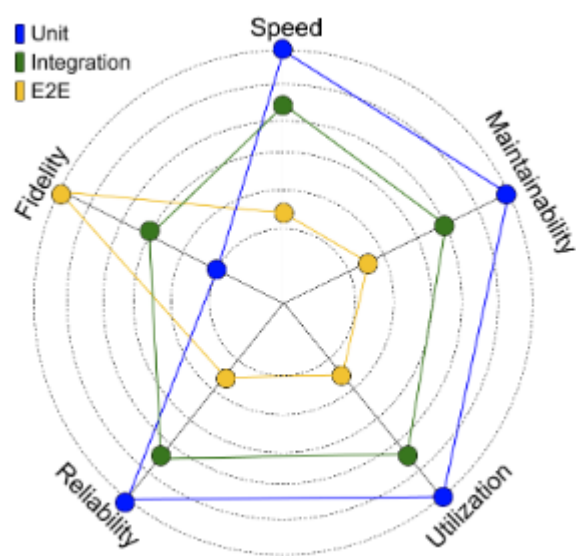


While useful, the test pyramid lacks the details you need as your test suite grows and you face challenging trade-offs. **To scale your test suite, go beyond the test pyramid.**

**The *SMURF* mnemonic is an easy way to remember the tradeoffs to consider when balancing your test suite:**

- **Speed:** Unit tests are faster than other test types and can be run more often—you'll catch problems sooner.
- **Maintainability:** The aggregated cost of debugging and maintaining tests (of all types) adds up quickly. A larger system under test has more code, and thus greater exposure to dependency churn and requirement drift which, in turn, creates more maintenance work.
- **Utilization:** Tests that use fewer resources (memory, disk, CPU) cost less to run. A good test suite optimizes resource utilization so that it does not grow super-linearly with the number of tests. Unit tests usually have better utilization characteristics, often because they use test doubles or only involve limited parts of a system.
- **Reliability:** Reliable tests only fail when an actual problem has been discovered. Sorting through flaky tests for problems wastes developer time and costs resources in rerunning the tests. As the size of a system and its corresponding tests grow, non-determinism (and thus, flakiness) creeps in, and your test suite is more likely to become unreliable.
- **Fidelity:** High-fidelity tests come closer to approximating real operating conditions (e.g., real databases or traffic loads) and better predict the

behavior of our production systems. Integration and end-to-end tests can better reflect realistic conditions, while unit tests have to simulate the environment, which can lead to drift between test expectations and reality.



A [radar chart](#) of Test Type vs. Test Property (i.e. SMURF).  
Farther from center is better.

**In many cases, the relationships between the SMURF dimensions are in tension:** improving one dimension can affect the others. However, if you can improve one or more dimensions of a test without harming the others, then you should do so. When thinking about the types of your tests (unit, integration, end-to-end), your choices have meaningful implications for your test suite’s cost and the value it provides.



Labels: Adam Bender , TotT

