A Paradigm Clarification

A test framework is an experiment.

I want to reinforce this point: a test framework is an experiment.

I’m not going to reveal any deep secrets, but the combination of elements is non-obvious. The concepts are simple, and once you build one test framework in this pattern, you can build subsequent versions very quickly.

Premature concretization is your enemy, and optimization is not your friend, *yet*.

A Paradigm Clarification

Developers test differently than specialized testers; we use the same terms, but they mean different things. We also have different aims.

This talk shows how to build a test framework that is external to the code base, and that is focused on deep and broad functional testing. No mocks and no priority on fast-running tests for a CI/CD pipeline. Built correctly, this framework supports end-to-end testing.

Developers often fall prey to the biases of confirmation and congruence, because they are focused on verifying and validating their code to requirements. The specialist tester looks for the non-optimal behavior; this talk is about supporting this attitude in test automation.