What I *am* going to talk about

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.

What I *am* going to talk about

How to build a straightforward test automation framework that focuses on functional tests, and that can support end-to-end tests.

This kind of test framework is inherently custom, in that it is closely aligned to the applications under test.

I believe that a test framework should be ephemeral and temporary: the goal should be on learning about the applications under test and on the business logic, and should not be on building the perfect test framework. The goal is on improving the models of business process that support and drive your tests; as your models improve, build better, more effective test frameworks.