

Some things I'm *not* going
to talk about:

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.

Some things I'm *not* going to talk about:

- ✦ Test-Driven Development
- ✦ Behavior-Driven Development and testing, and the given-when-then syntax
- ✦ The reasons why test automation is important
- ✦ The dialectic between test automation and exploratory investigation by human beings
- ✦ The “checking” vs “testing” labeling dispute
- ✦ The various full-featured, do-everything, test automation juggernauts-in-a-box like Tricentis, HP Quality Center, Katalon Studio, etc.