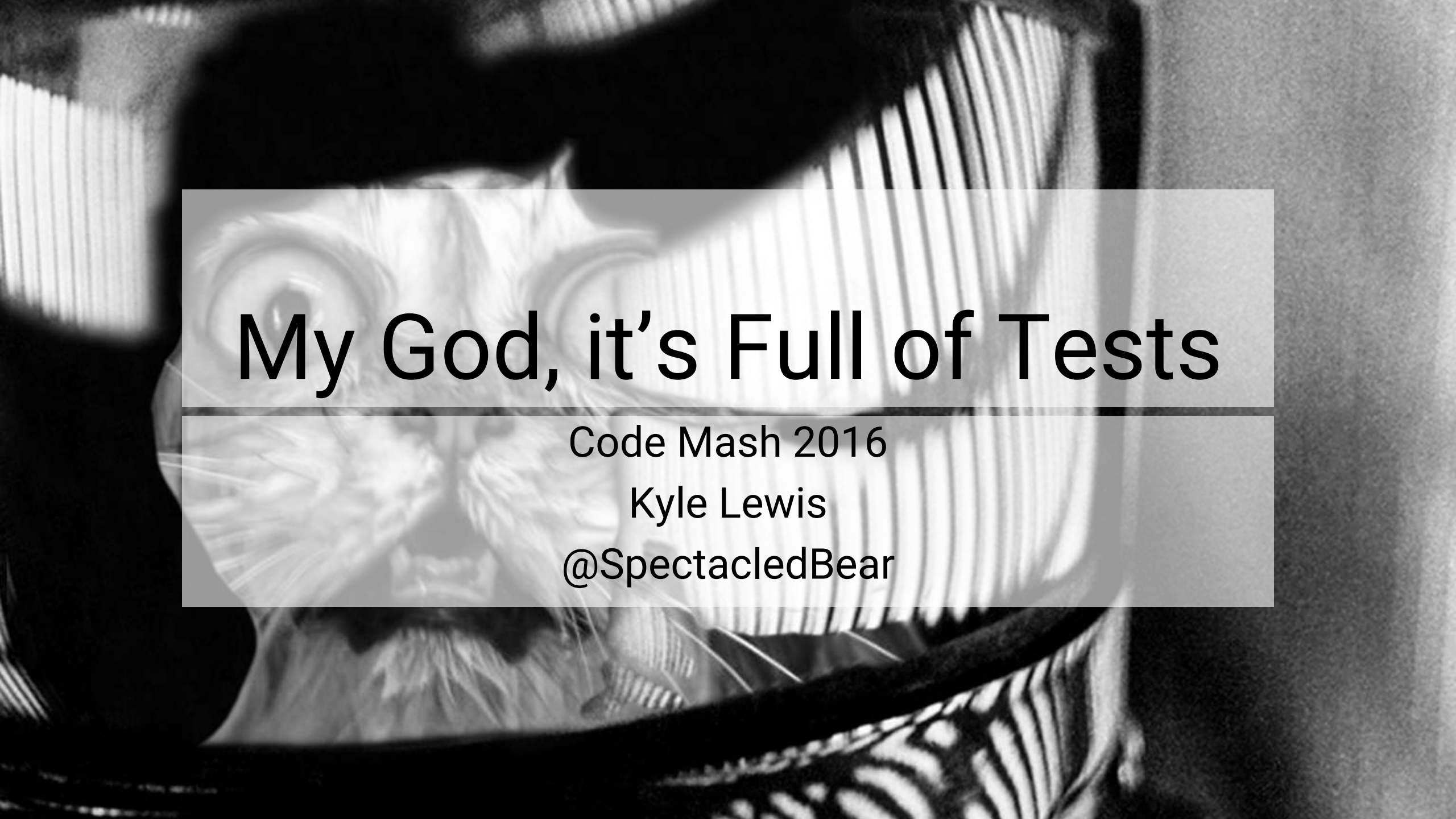


My God, it's Full of Tests

Code Mash 2016

Kyle Lewis

@SpectacledBear

Outline

- Context
- **Test models**
- Questions

Context

Rule One: Automation may not injure a human being or, through inaction, allow a human being to come to harm.

Who Is This Guy?

MAGNET FORENSICS®

Continuous Delivery

Driving Cars in a Traffic Jam by epSos.de
Licensed under a CC BY 2.0 license
https://flic.kr/p/9w8eWL

DevOps

- Automation in all things
  - Infrastructure
  - System tests
  - Monitoring
- Incremental adoption
  - Systems Thinking
  - Amplify Feedback Loops
  - Culture of Continual Experimentation and Learning

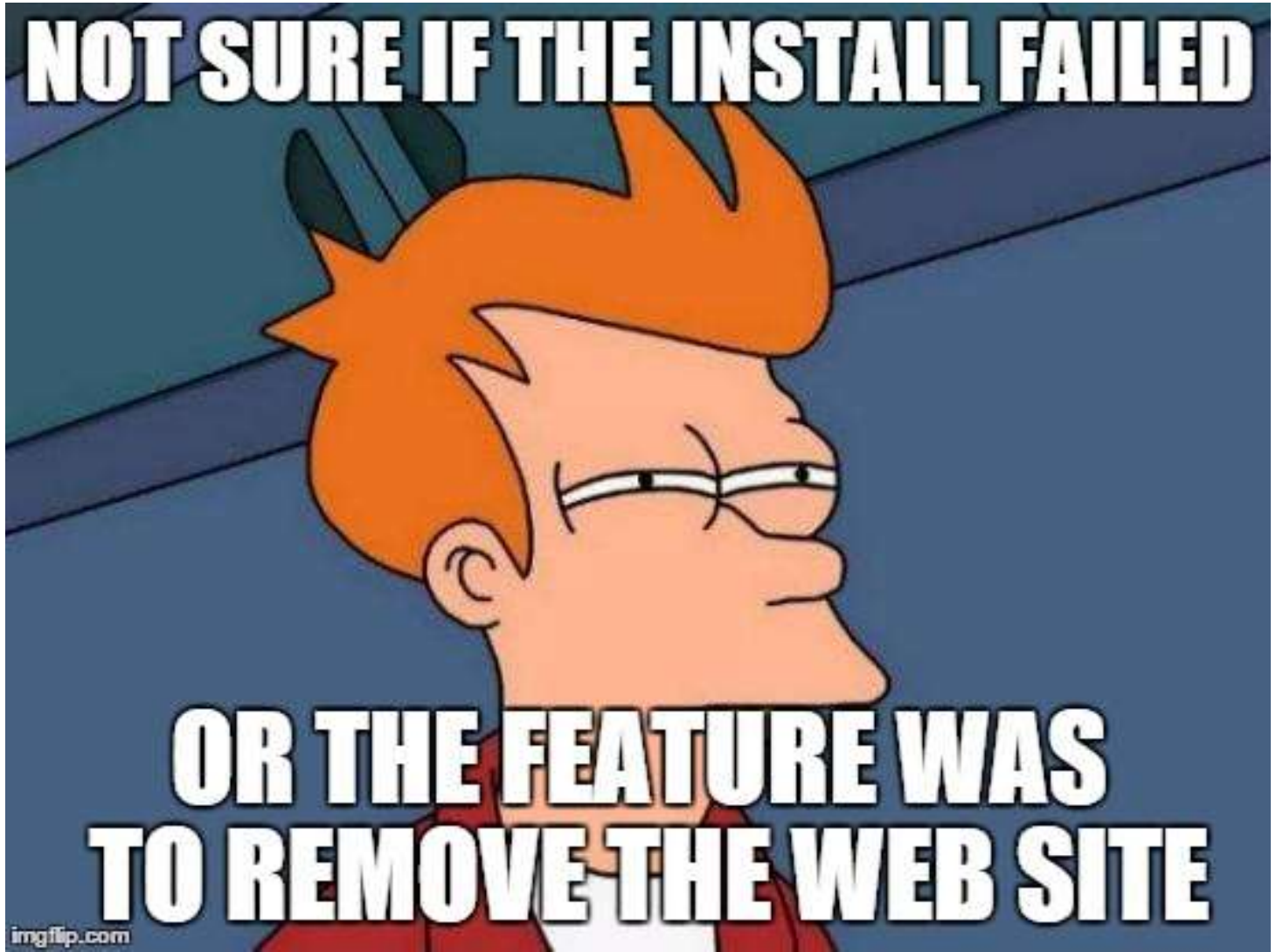
Test Models

Rule Two: Automation must obey the orders given it by human beings except where such orders would conflict with the First Law.

Sandbox Introduction

Butterfly Catcher by Jeff Hitchcock
Licensed under a CC BY 2.0 license
https://flic.kr/p/dBG25G

Build Manifest Tests



Build Manifest Tests Demo

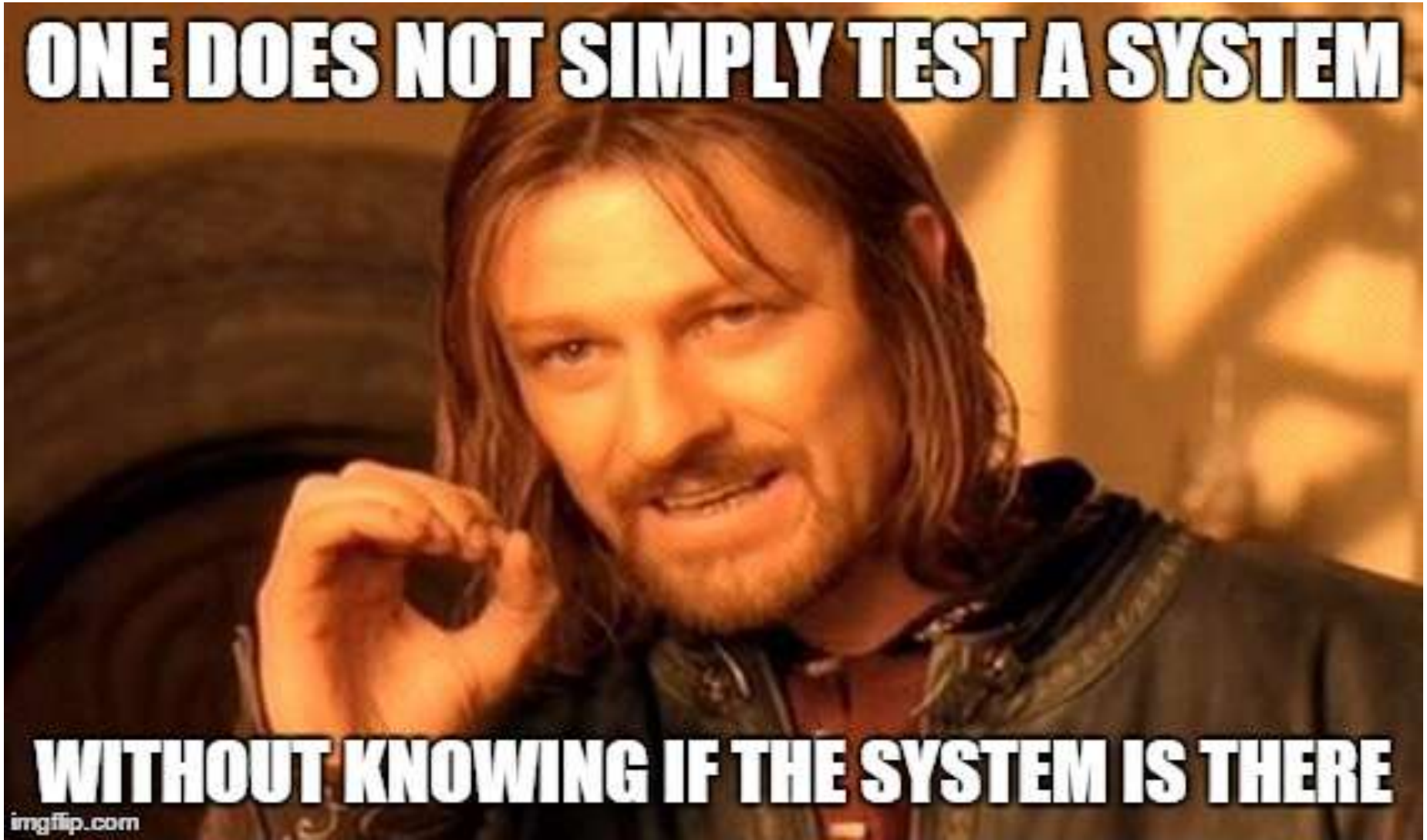
Is this even what we expect?

Build Manifest Tests

A model for comparing what we built to what was deployed.

```
"Manifest": [
    {
        "File": "SpectacledBear.CodeMash2016.WebApi.dll",
        "Version": "1.0.0.0",
        "Checksum": "24B381CFEBF3BC96C300676717E90F7B“
    },
    ...
]
```

Build Validation Tests



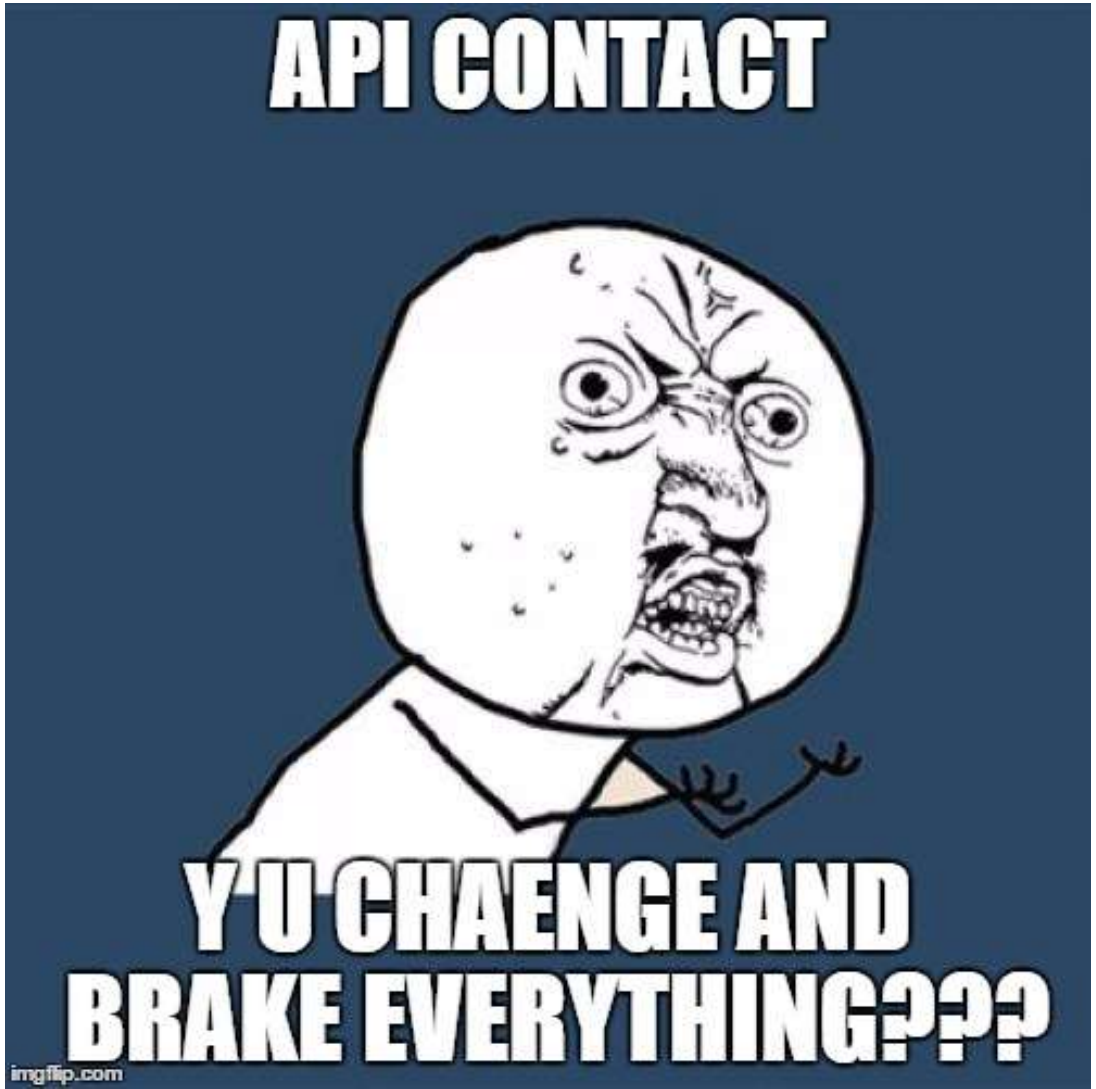
Build Validation Tests Demo

Is the product even there?

Build Validation Tests

A model for checking an environment prior to executing other automated tests.

Contract Tests



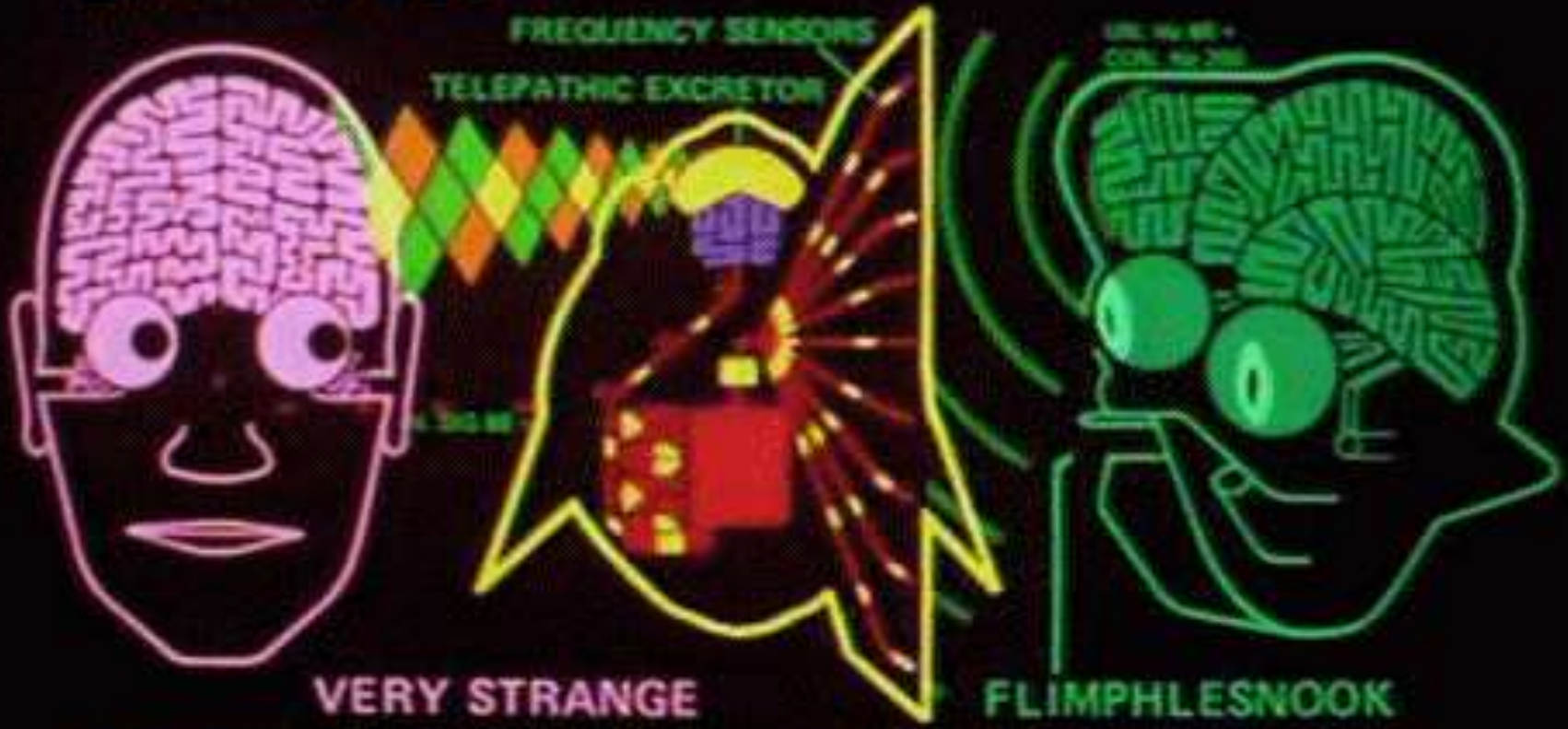
Contract Tests Demo

How do we test our boundaries earlier?

Contract Tests

A model for checking a consumer-service contract in isolation from each other.

BABEL FISH



UNCONSCIOUS FREQUENCIES AND THEN EXCRETING TELEPATHICALLY A MATRIX FORMED FROM THE CONSCIOUS FREQUENCIES AND NERVE SIGNALS PICKED UP FROM THE SPEECH CENTRES OF THE BRAIN.

Domain Language Tests Demo

Why do we stop using our language when we test our system?

Domain Language Tests

A model for documenting specifications as tests.

Monitoring Tests

Monitoring Tests Demo

How do we keep up with all of these moving pieces?

Monitoring Tests

A model for monitoring a system for evaluation and comparison.


```
{
        "Result“ : "pass",
        "QueryResponseTime“ : 13,
        "TotalChanges“ : 2,
        "SqliteVersion“ : "3.8.11.1",
        "Tables“ : ["Hobbits"]
}
```

Continuous Improvement

- What are your assumptions and behaviours that are not being checked?
- Experiment and iterate!
- Think more on the five models presented:
  - Manifest checking
  - Build validation
  - Service contract validation
  - Domain specifications
  - Health monitoring
- Go forth and learn!

Questions?

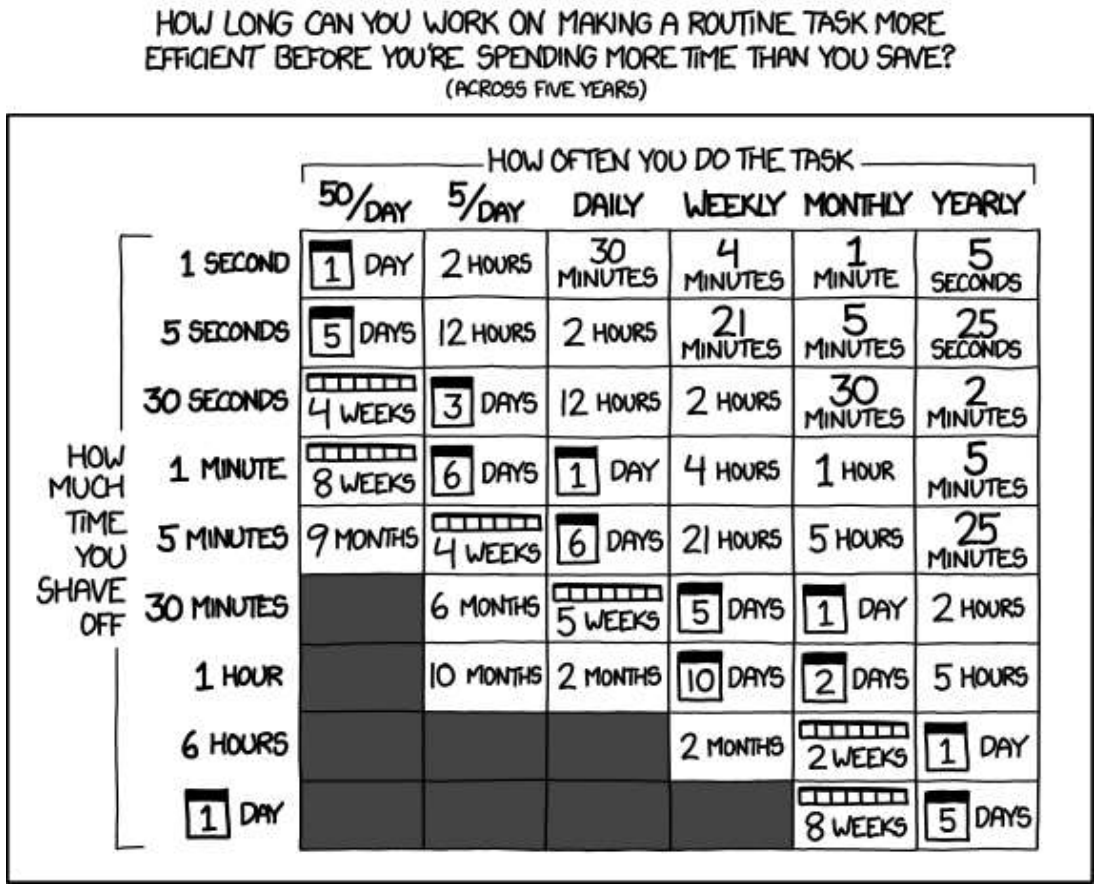
Rule Three: Automation must protect its own existence as long as such protection does not conflict with the First or Second Laws.

Where To Find Me/This Talk

SpectacledBear
on Twitter and GitHub

thekyle@gmail.com

https://github.com/SpectacledBear/codemash-2016



https://xkcd.com/1205/

Resources

- The Three Ways by Gene Kim http://itrevolution.com/the-three-ways-principles-underpinning-devops/
- Health Endpoint Monitoring Pattern https://msdn.microsoft.com/en-us/library/dn589789.aspx
- PactNet test framework https://github.com/SEEK-Jobs/pact-net
- Machine.Specifications https://github.com/machine/machine.specifications