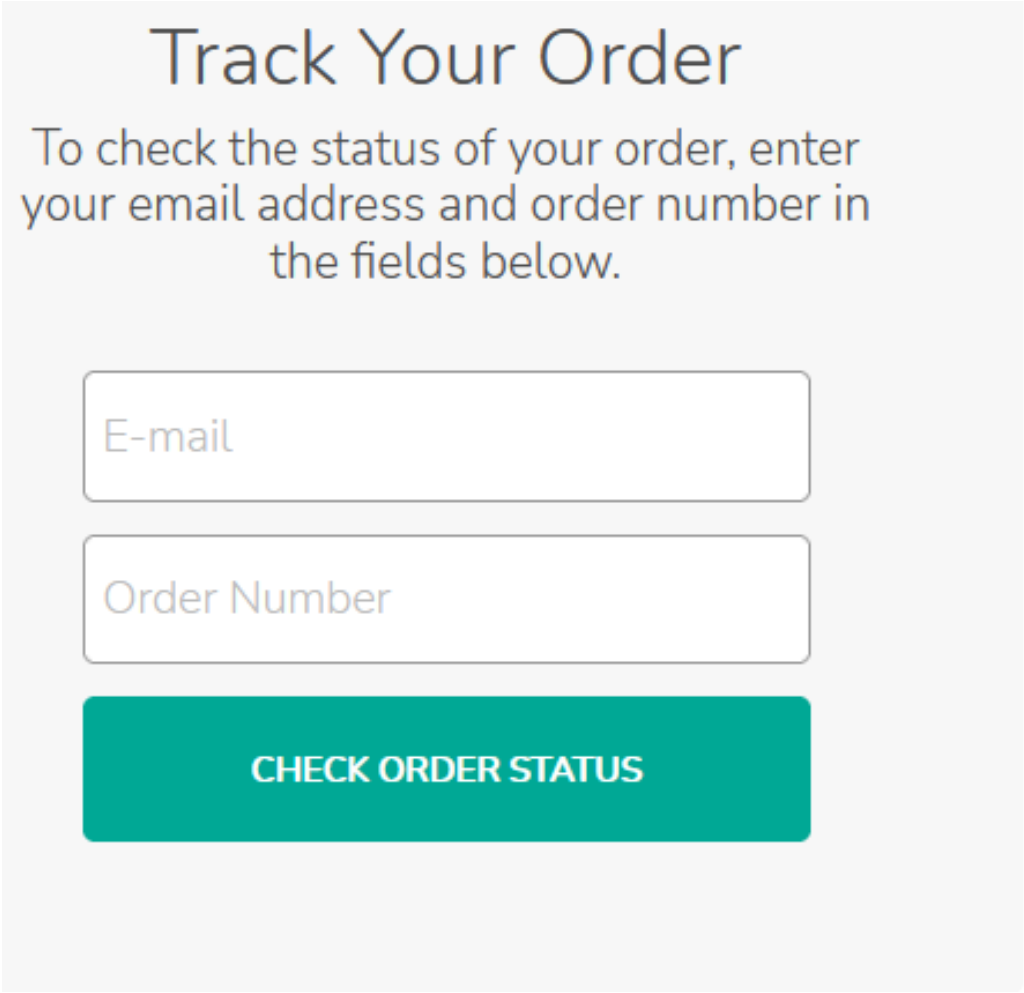
T-E-S-T is not a four letter word!

Email address valid?

Order number numeric?

Order not found?

Valid entries for email and order number
Interaction with data to retrieve order
On screen result



Outline

What is unit testing?

Types of testing

Why you should write tests

Test examples and demo

Cost of testing

Strategies and getting started

About me

Mark Kalal

Software development / technology solutions

mdkalal@gmail.com, mark.kalal@wellsfargo.com

@MarkKalal

https://www.linkedin.com/in/markkalal/

What is Unit testing?

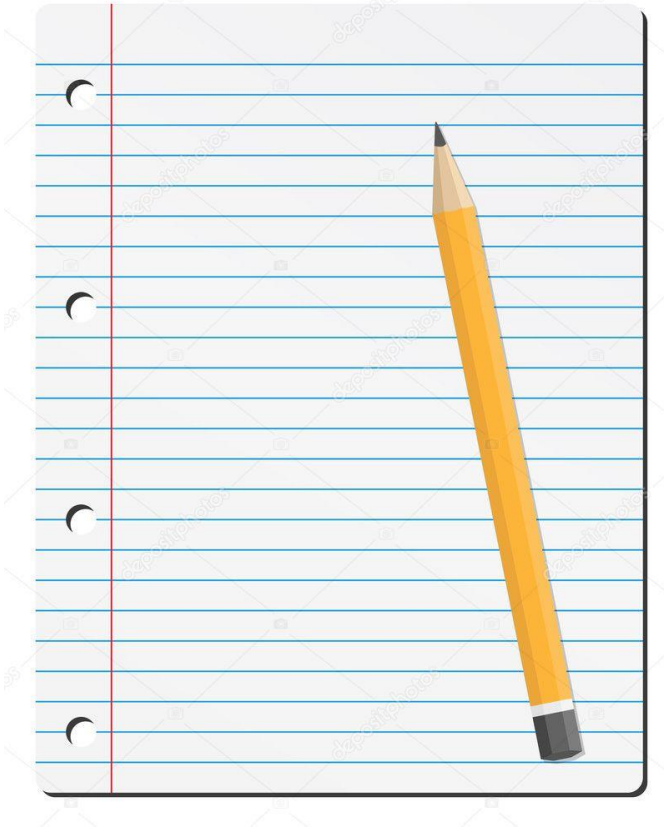
Testing the smallest bits

Writing tests

Writing tests

Writing code to test code

Automated vs manual

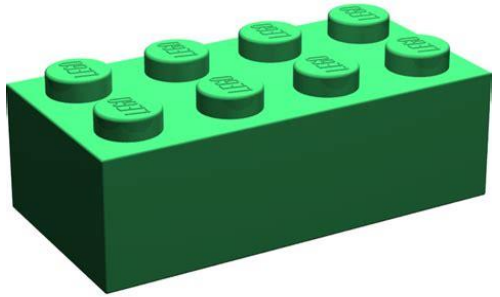


Types of Testing

Unit testing

Integration testing

End-to-end testing

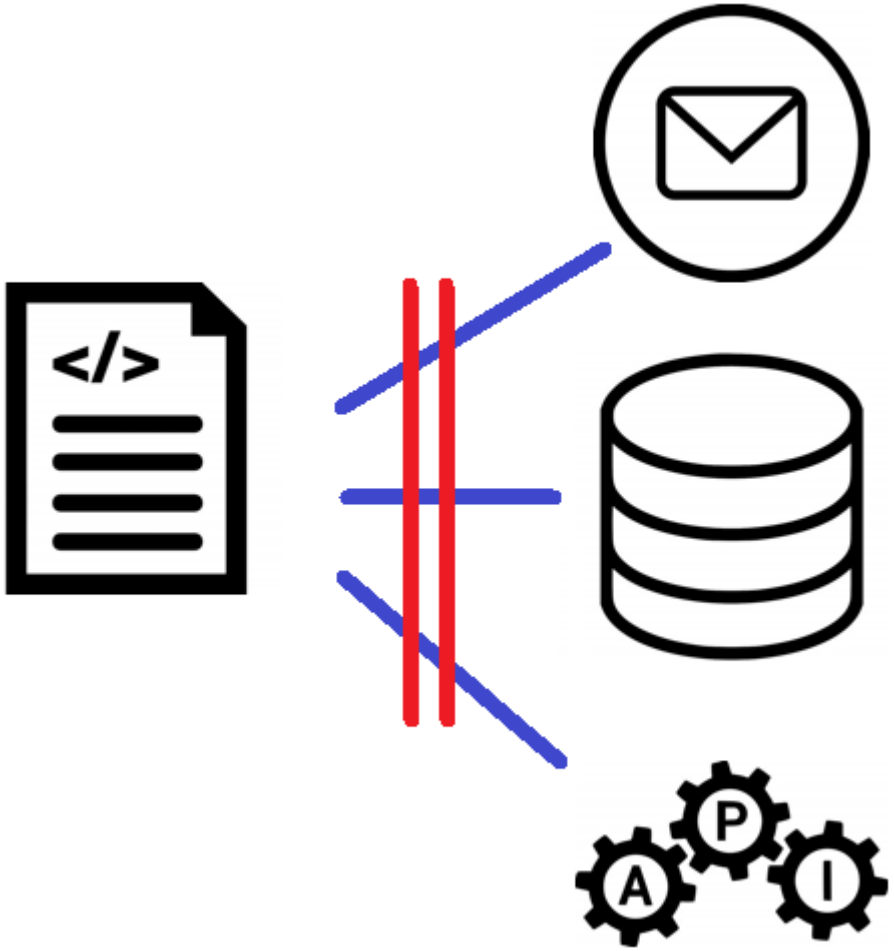


Unit Testing

Fast to run, repeatable

Efficient

Don’t give complete view



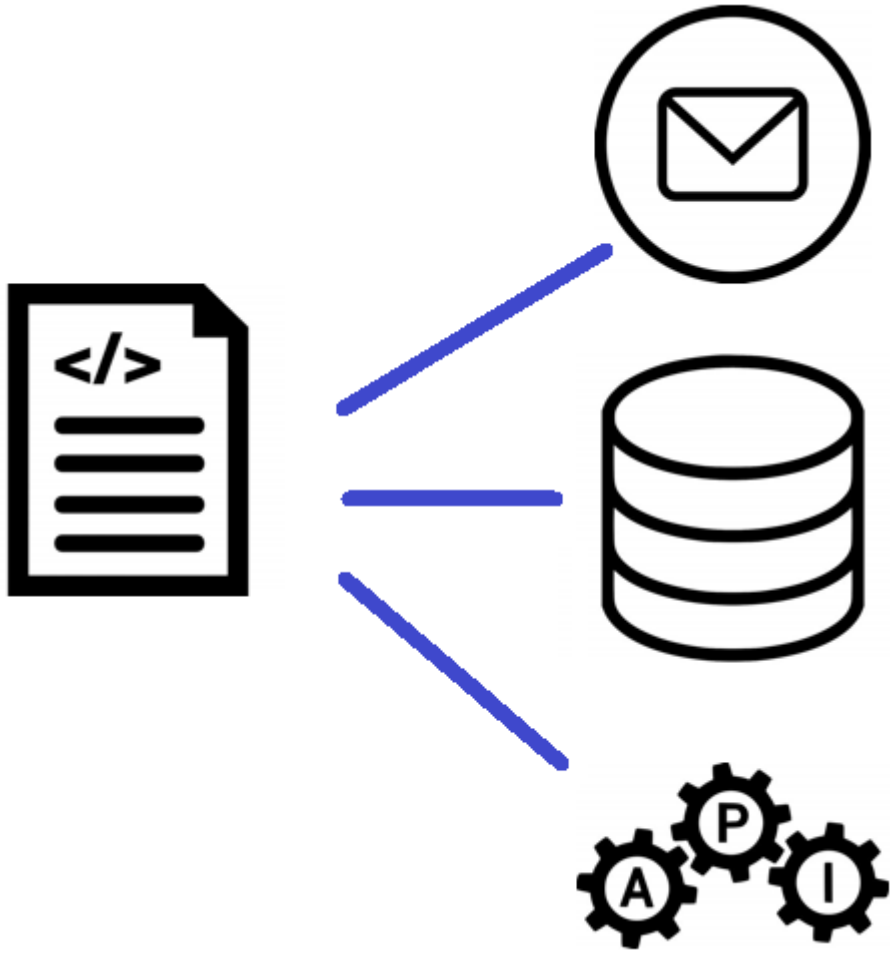
https://twitter.com/i/status/813056723271450624

Integration Testing

Greater confidence

Impractical for rapid iterations

More prone to failure

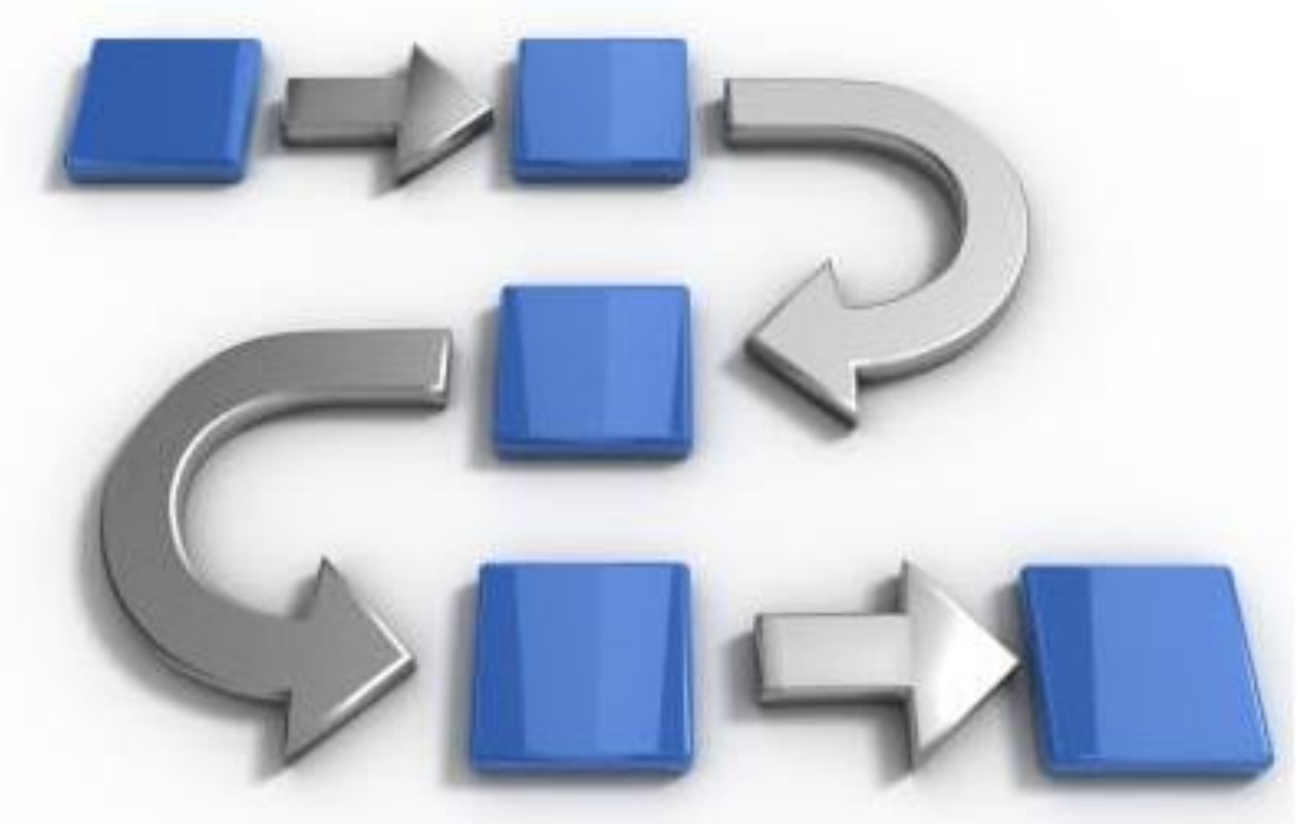


End To End Testing

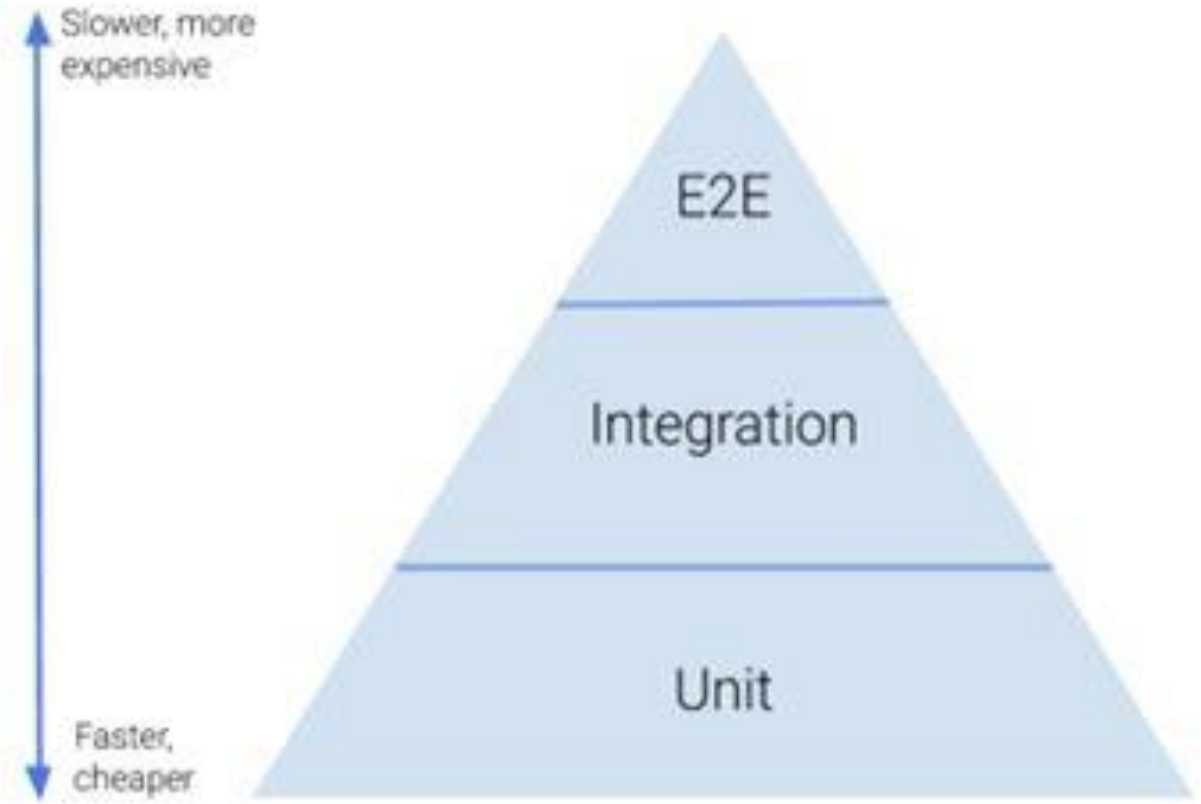
Greatest confidence

Longest, hardest

“Brittle” to automate



Test Pyramid



https://betterprogramming.pub/the-test-pyramid-80d77535573

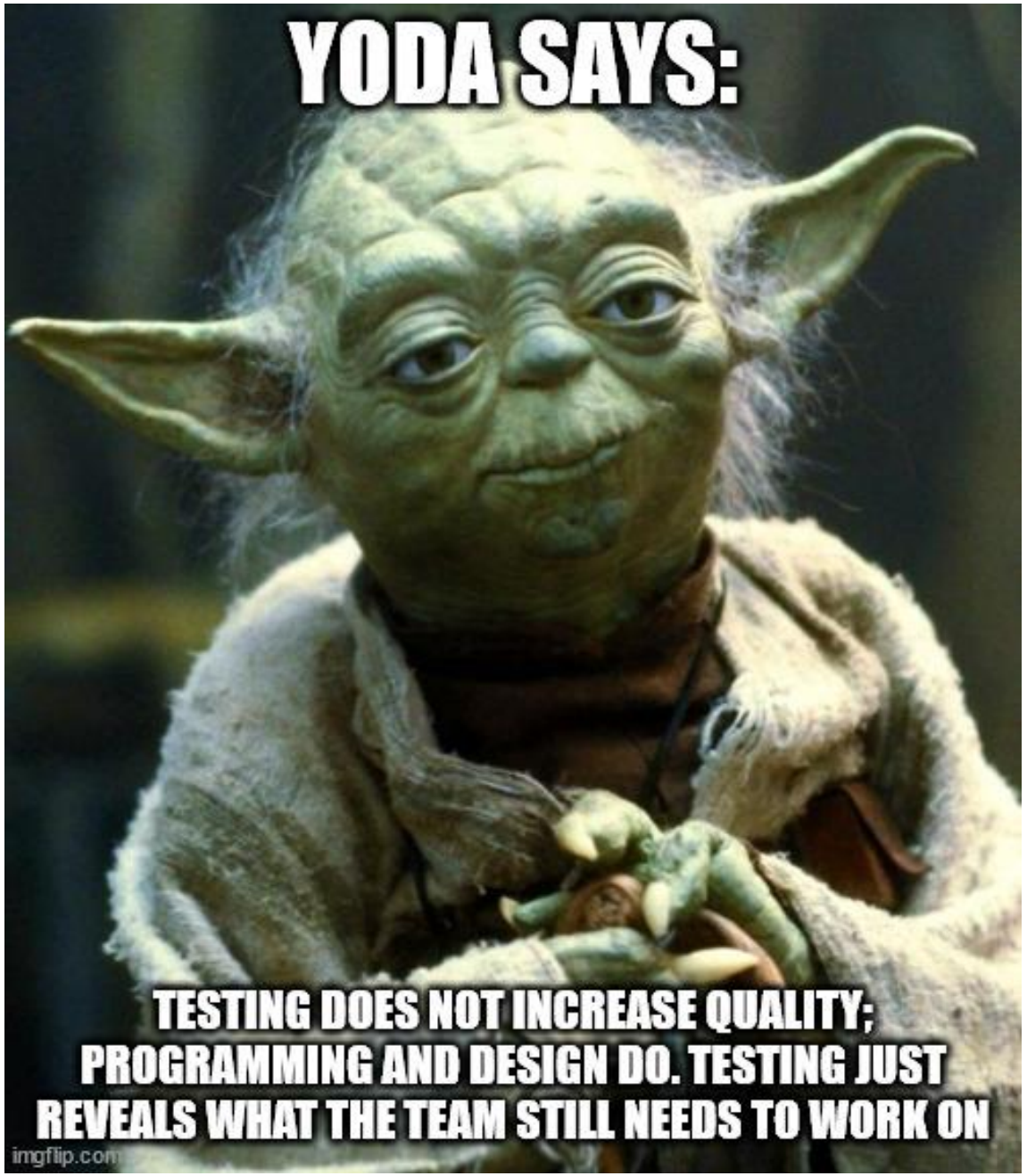
Benefits

Easy testing early and often

Develop with confidence

Refactor with confidence

Better tests = better code = higher quality = better product = happier users


YODA SAYS:
TESTING DOES NOT INCREASE QUALITY;
PROGRAMMING AND DESIGN DO. TESTING JUST
REVEALS WHAT THE TEAM STILL NEEDS TO WORK ON
imgflip.com

Demo

Framework

Test Organization (AAA)

Naming

Coverage

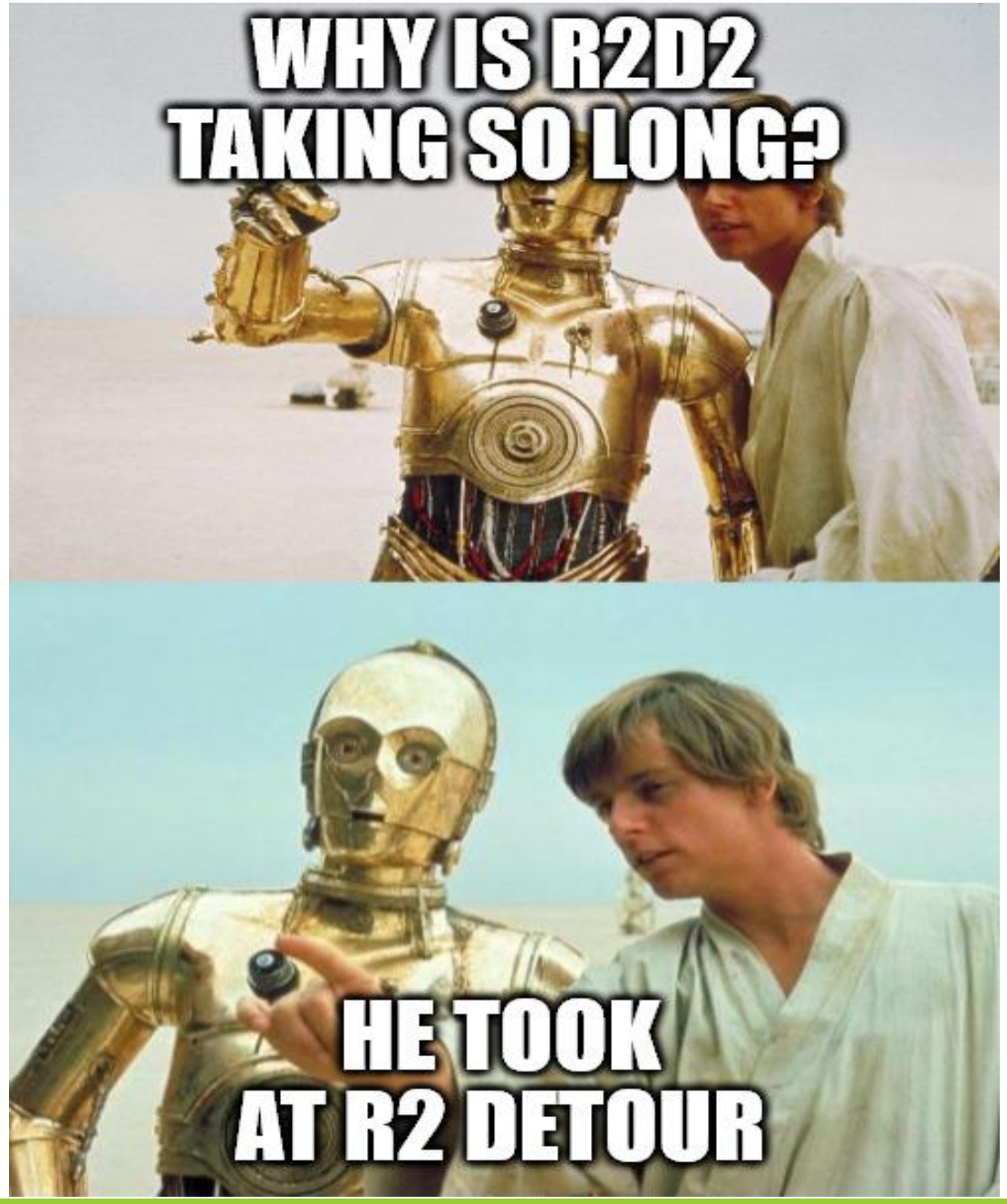
Refactoring

Fakes/Mocks

Stunt doubles

Mimic state/behavior

“Remove” dependencies


WHY IS R2D2
TAKING SO LONG?
HE TOOK
AT R2 DETOUR

Testing can come with a price …

Adds initial time

Adds maintenance – every test you write is code that must be maintained

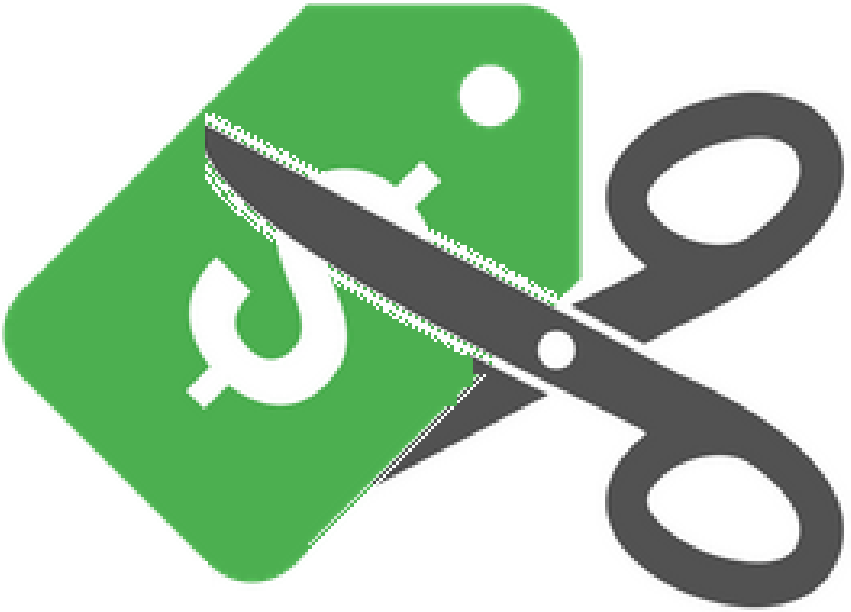
The code is messy and tests are hard

Or does it?

Adds initial time - Think of tests as design tools.

Adds maintenance – every test you write is code that must be maintained – Prioritize, focus on riskier scenarios

The code is messy and unit tests are hard – Consider “courser” tests, or integration tests



Strategies

Best practices

Easier for “Greenfield” development?

What about “Legacy” applications?

Best Practices

Organization (AAA, Naming convention)

“Stateless”

Precision

KISS

Write tests first (TDD)

“Legacy” Applications

Start with what you can

Refactor or not?

As code is added/refactored, include tests

Trouble spots are good candidates for tests

Summary

- ✓What is unit testing?
- ✓Types of testing
- ✓Why you should write tests
- ✓Test examples and demo
- ✓Cost of testing
- ✓Strategies and getting started

For more information

URList - https://www.theurlist.com/kalal-test

Unit vs Integration testing - https://earthly.dev/blog/unit-vs-integration/

Best Practices - https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices

Good Testing - https://codeanit.medium.com/developers-guide-write-good-test-5e3e3cdec78e

Get Started with TDD - https://dzone.com/articles/get-started-with-test-driven-development-a-beginne

MOQ framework - https://www.youtube.com/watch?v=dZ2Psa_Bn2Q

This slide deck and code project – https://github.com/mdkalal/T-E-S-T

TECH Masters

www.TechMasters-TC.com

Tuesdays, 7:45-9:00 am, Improving Enterprises, Bloomington

Thank you!

Feedback welcome

Mark Kalal - mdkalal@gmail.com

@MarkKalal

https://www.linkedin.com/in/markkalal/