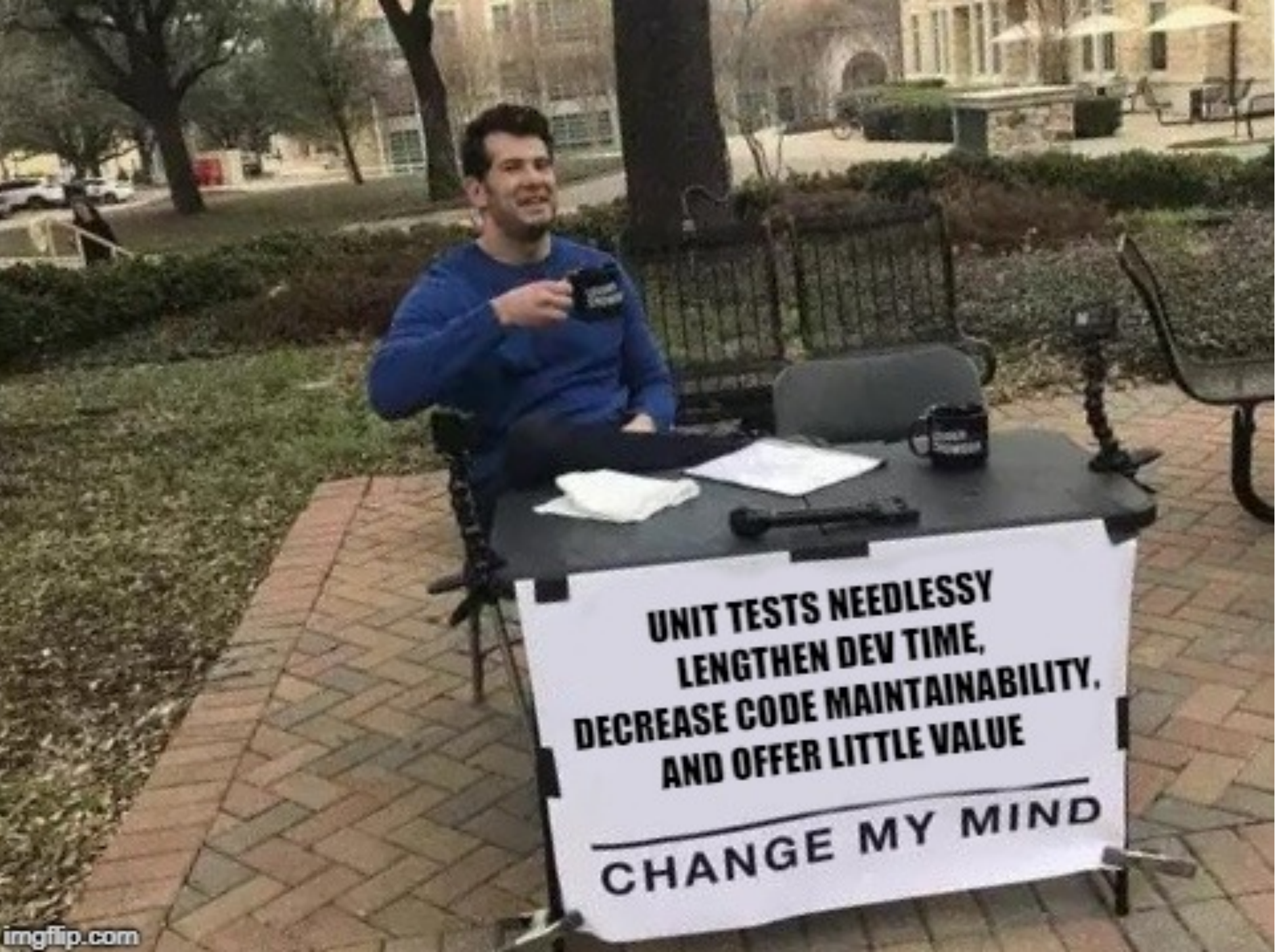
Fast, Robust, Accurate (WIP)

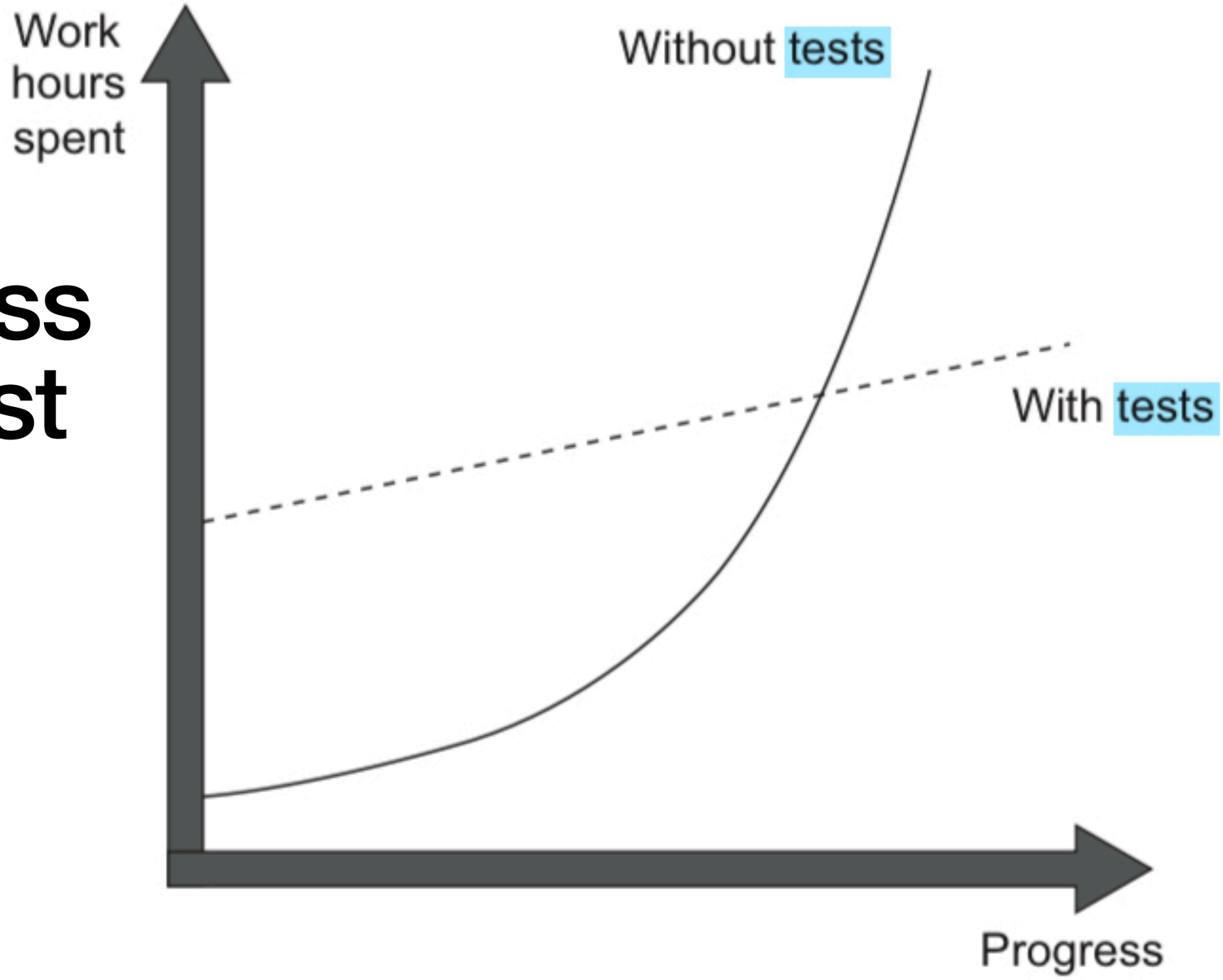
Or how to love your tests again

@xuapsdev

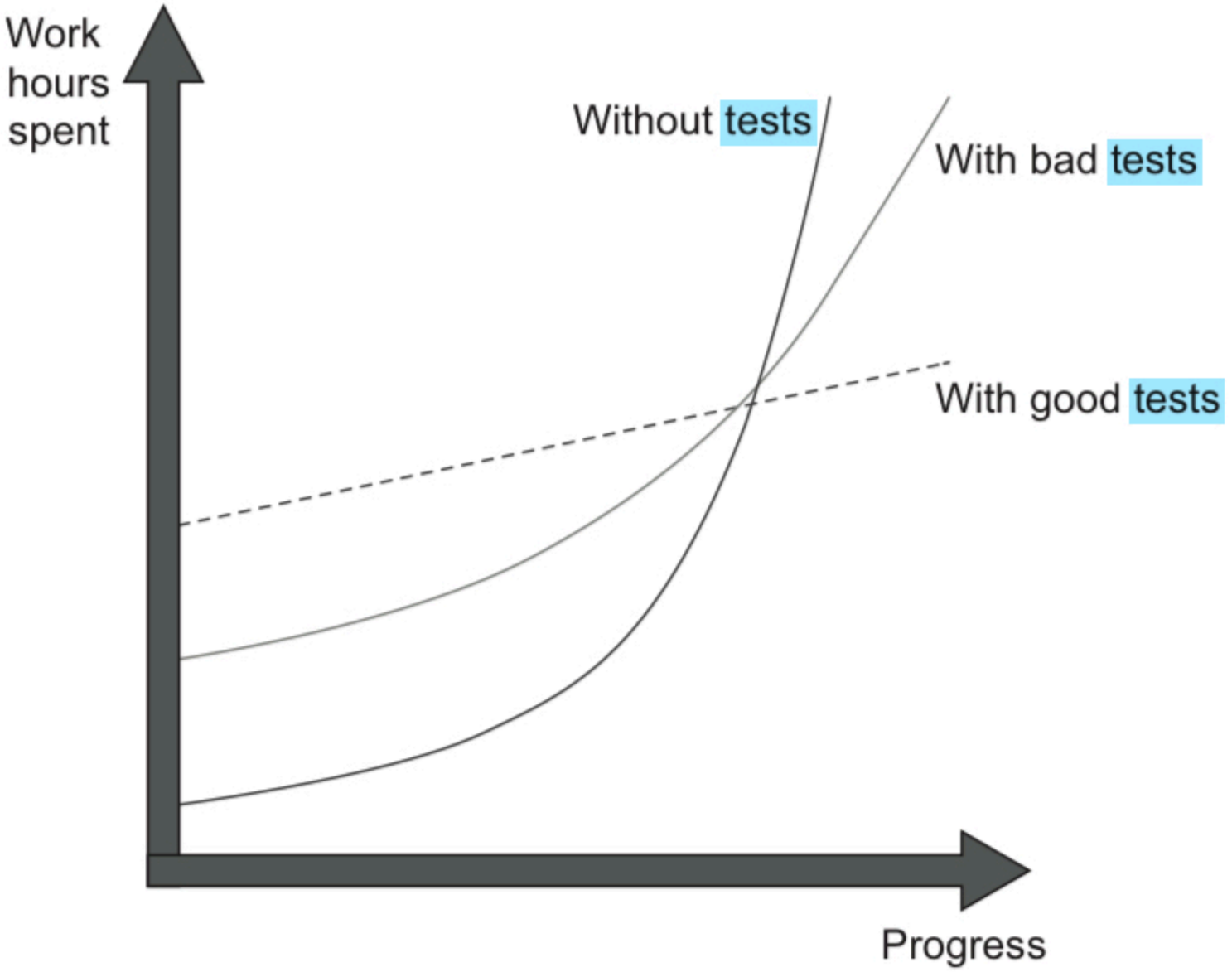


This is not a talk about opinions

1 - We progress faster with test

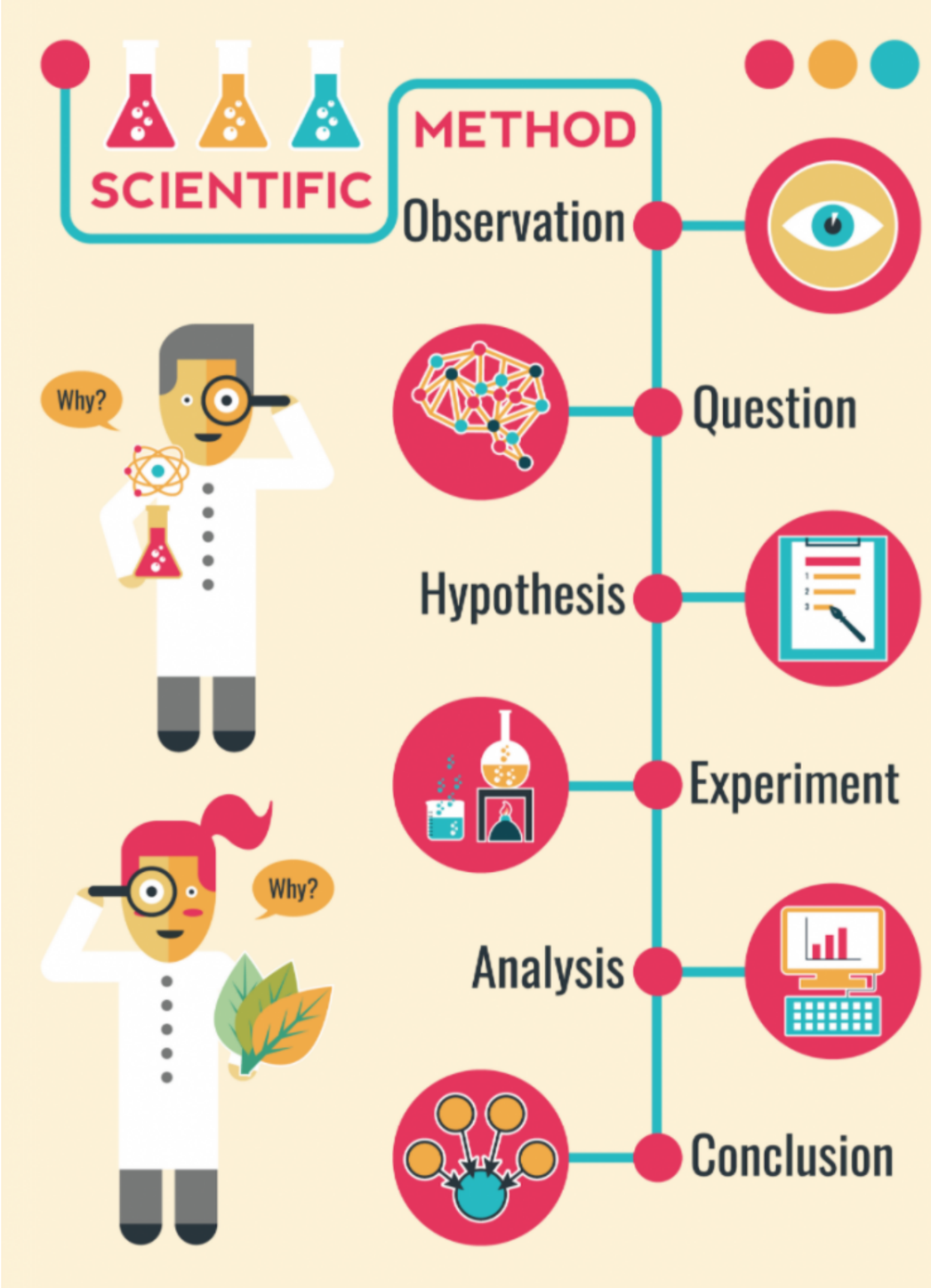


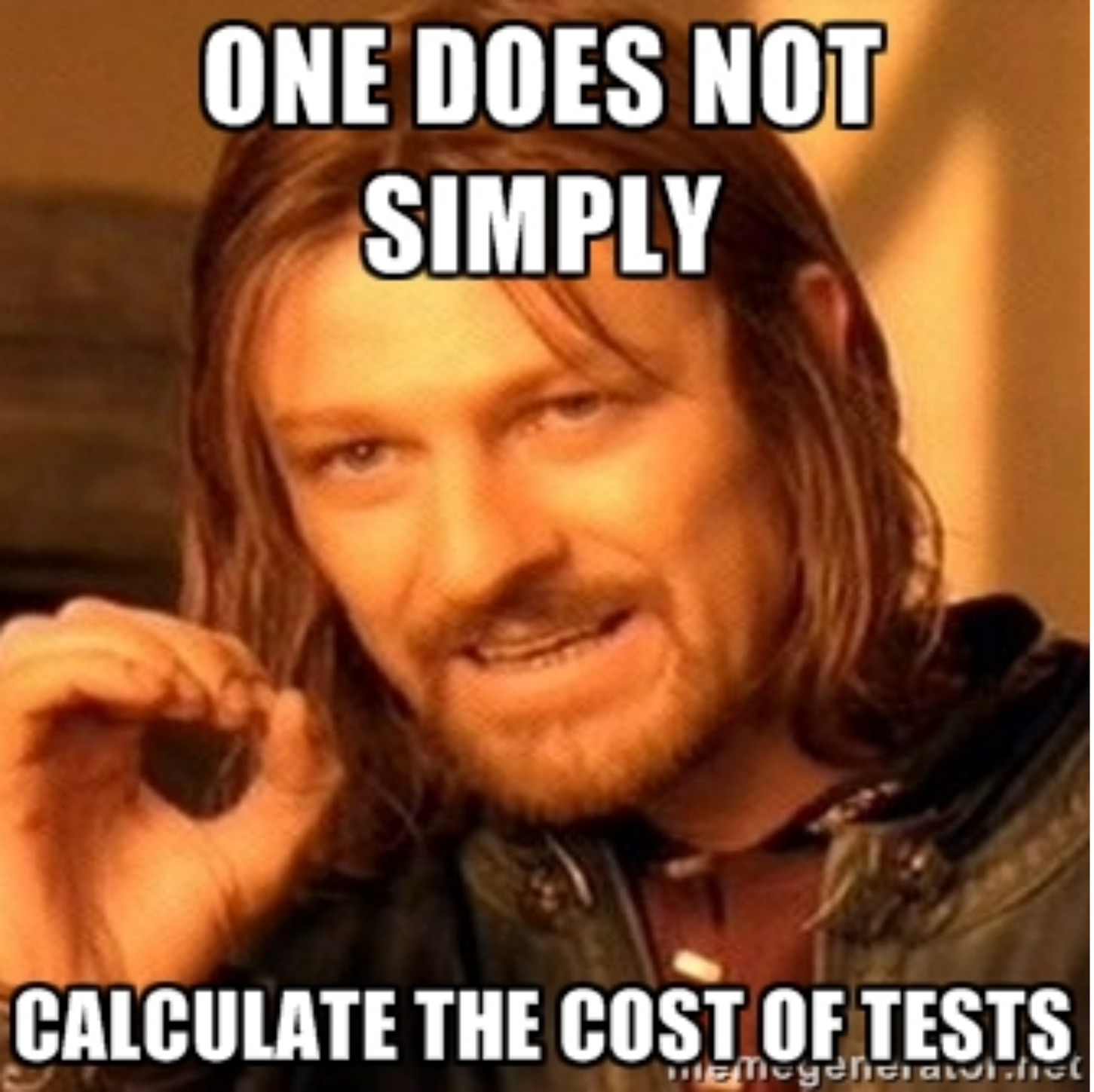
2 - Bad tests could be worst than no tests

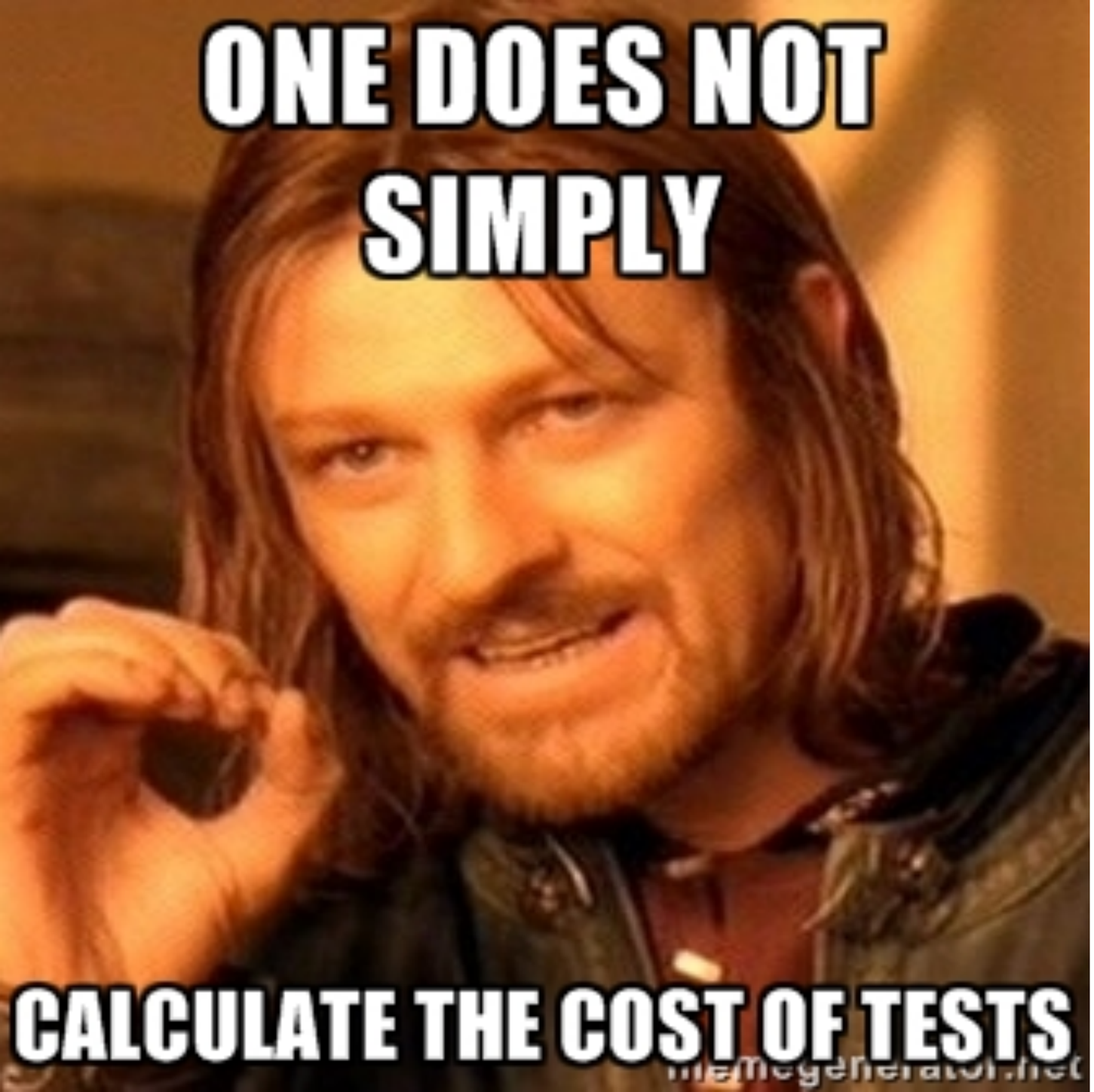



OPINIONS,
OPINIONS EVERYWHERE
makeameme.

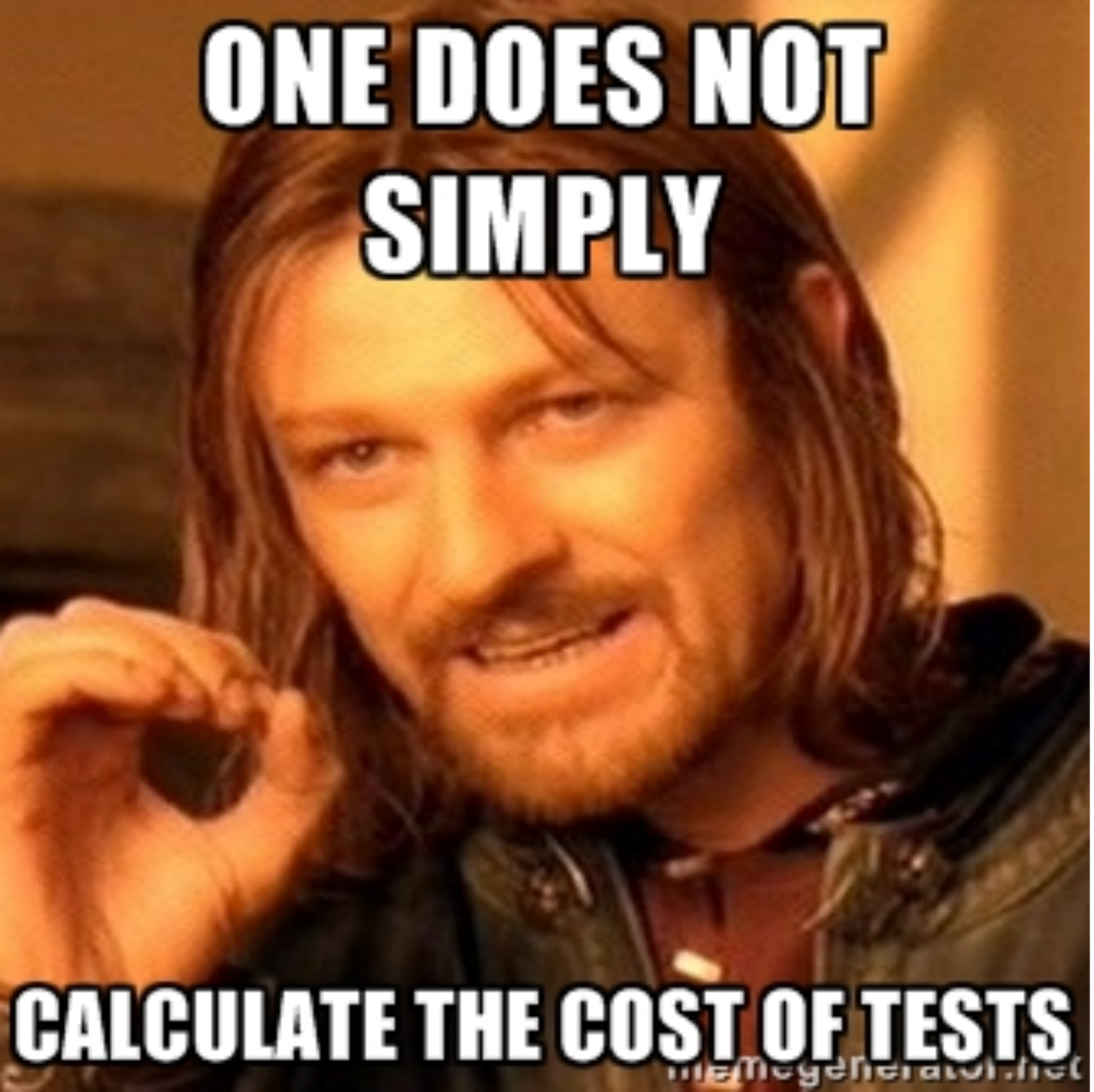
So, what is a bad test from an empirical point of view?




ONE DOES NOT
SIMPLY
CALCULATE THE COST OF TESTS

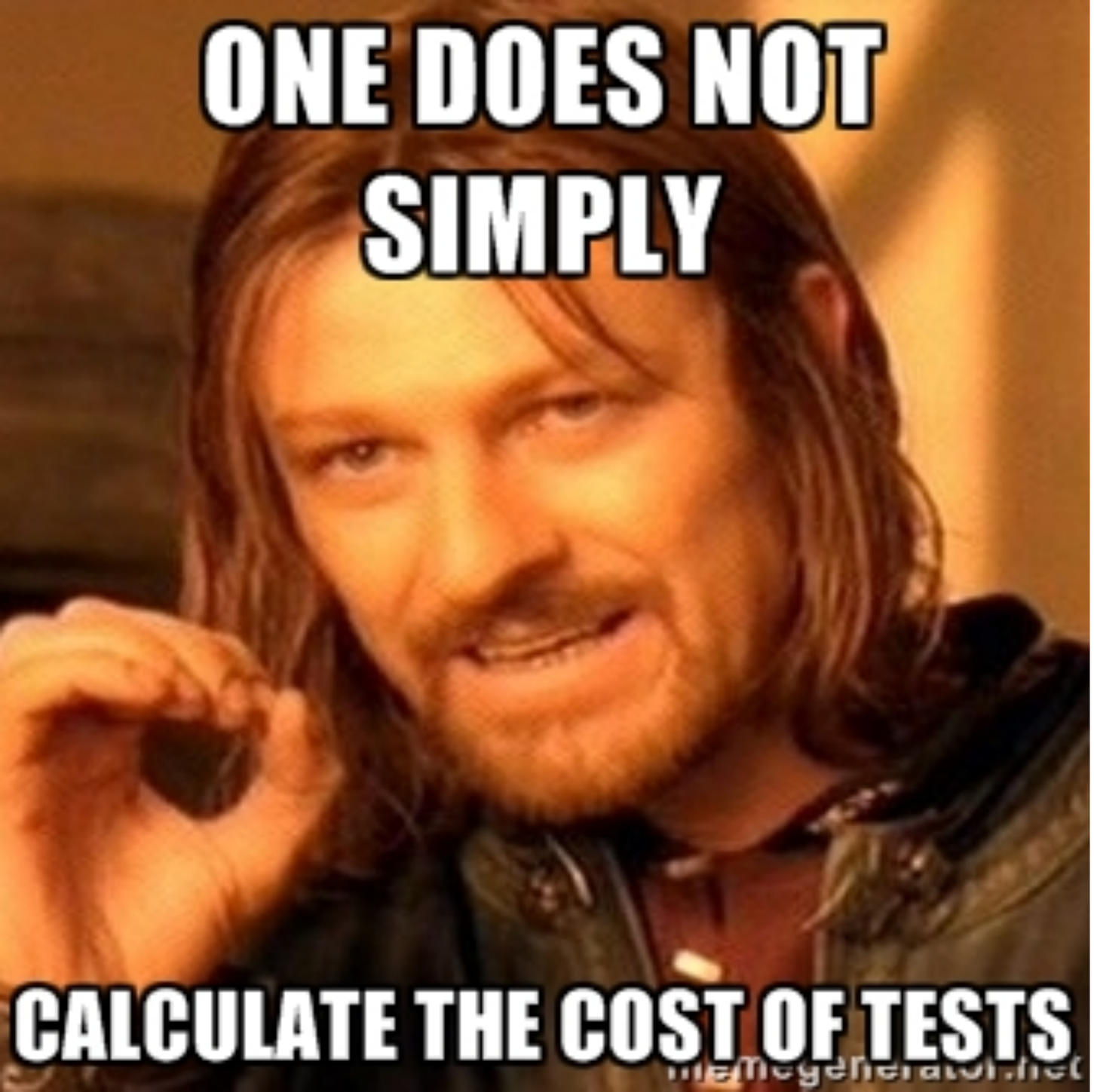


Slow tests



Slow tests

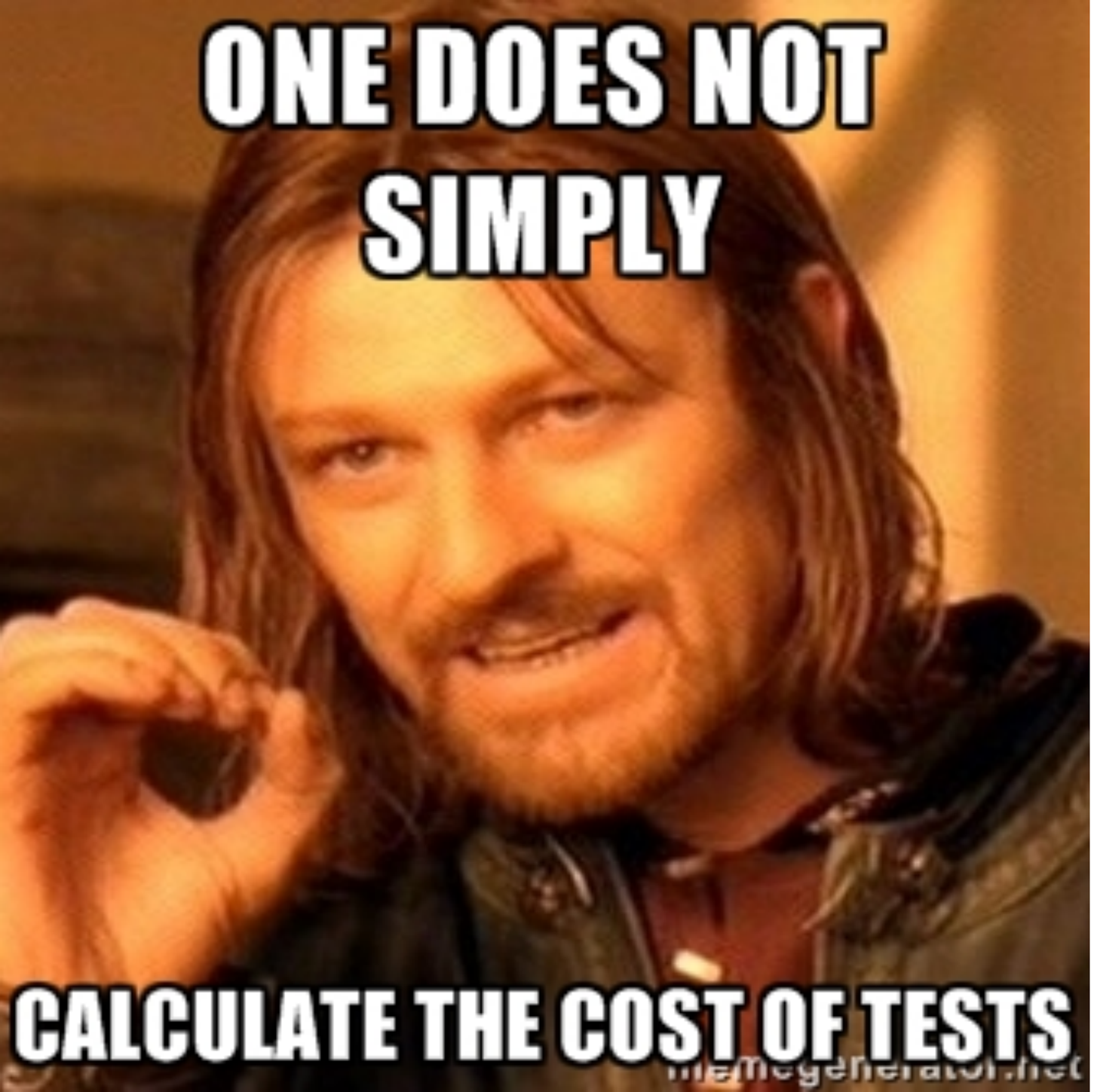
False alarms



Slow tests

False alarms

Difficult to read tests

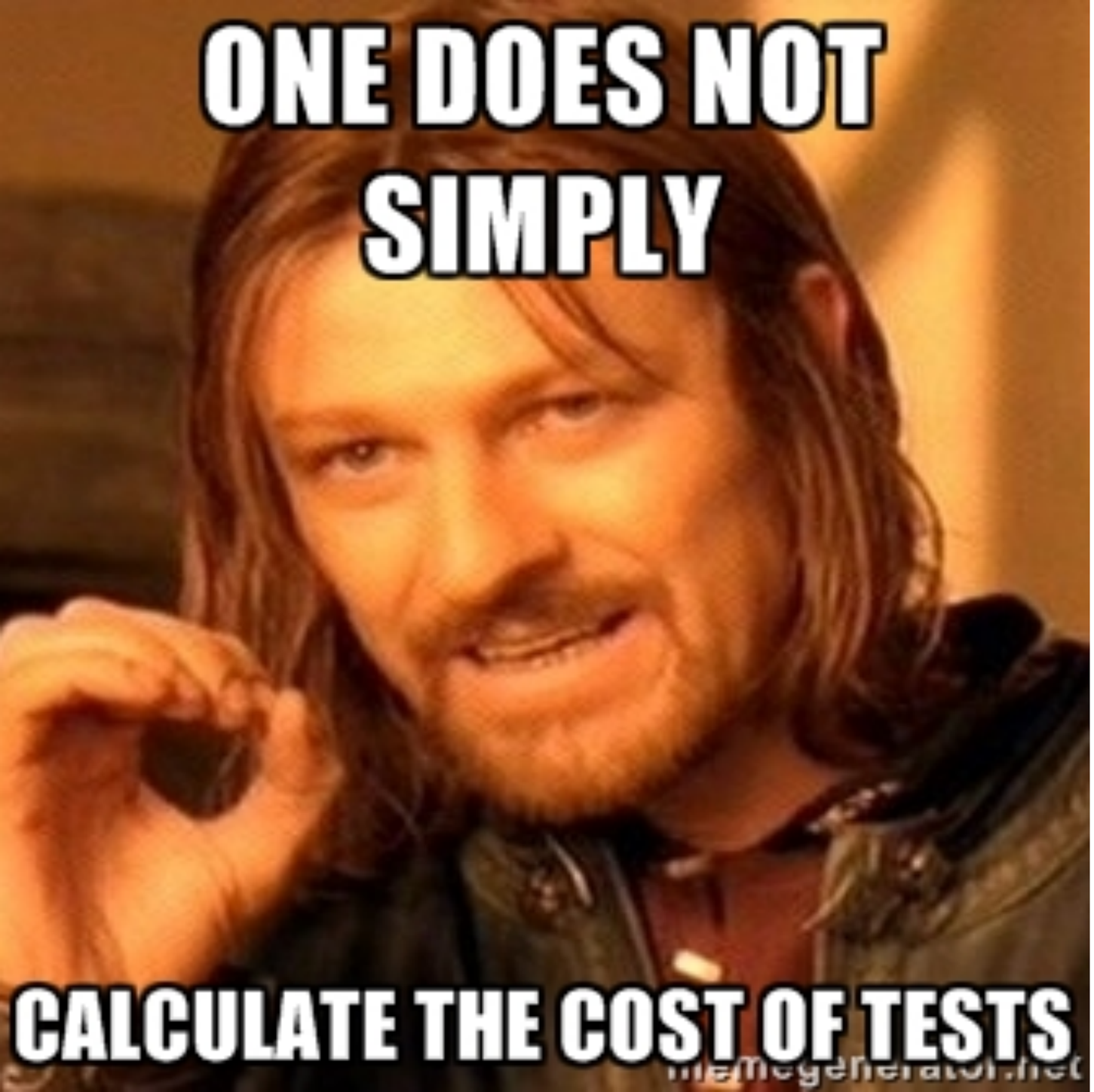


Slow tests

False alarms

Difficult to read tests

Brittle tests



Slow tests

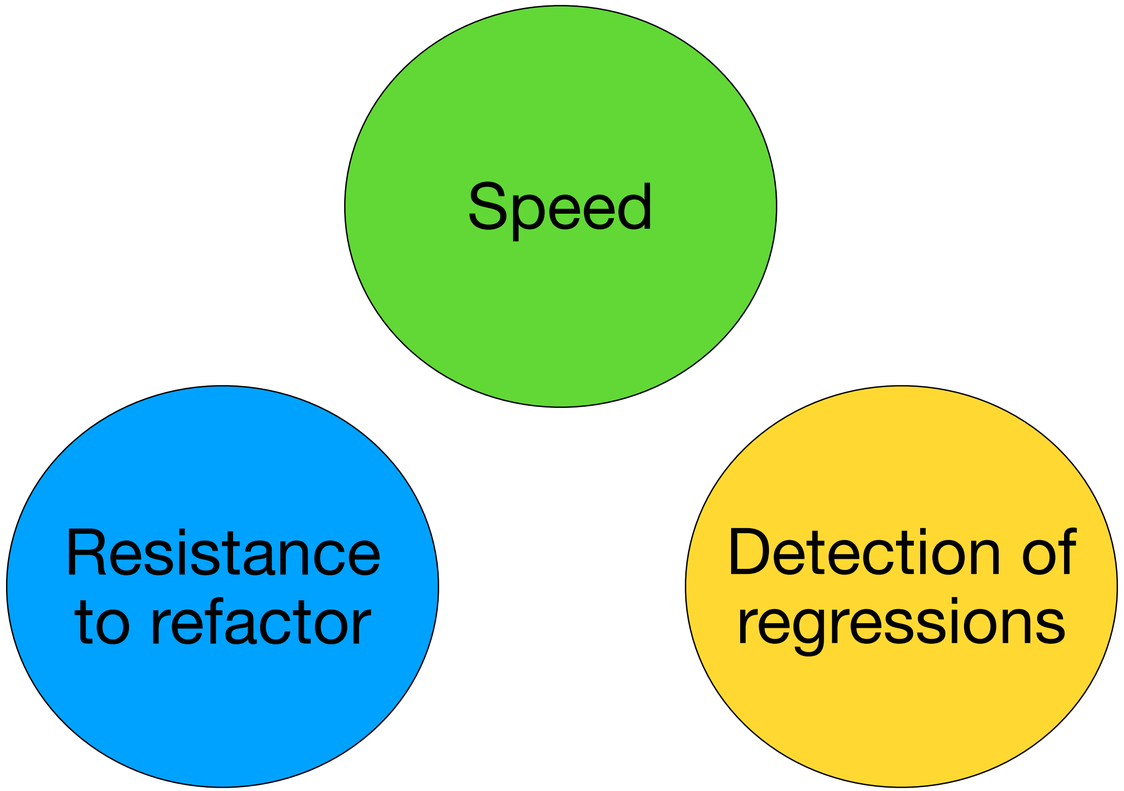
False alarms

Difficult to read tests

Brittle tests

What else?

3 properties

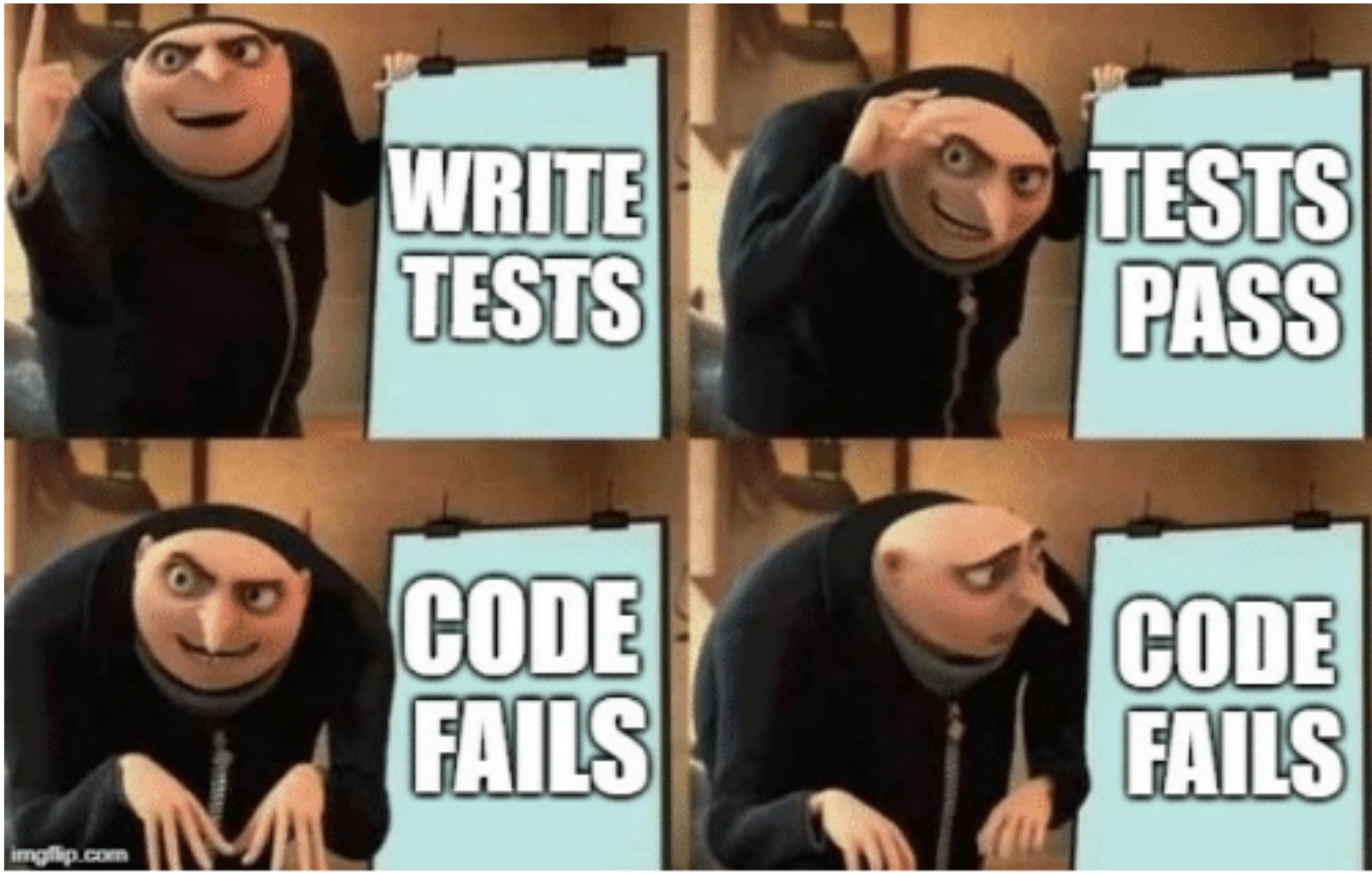


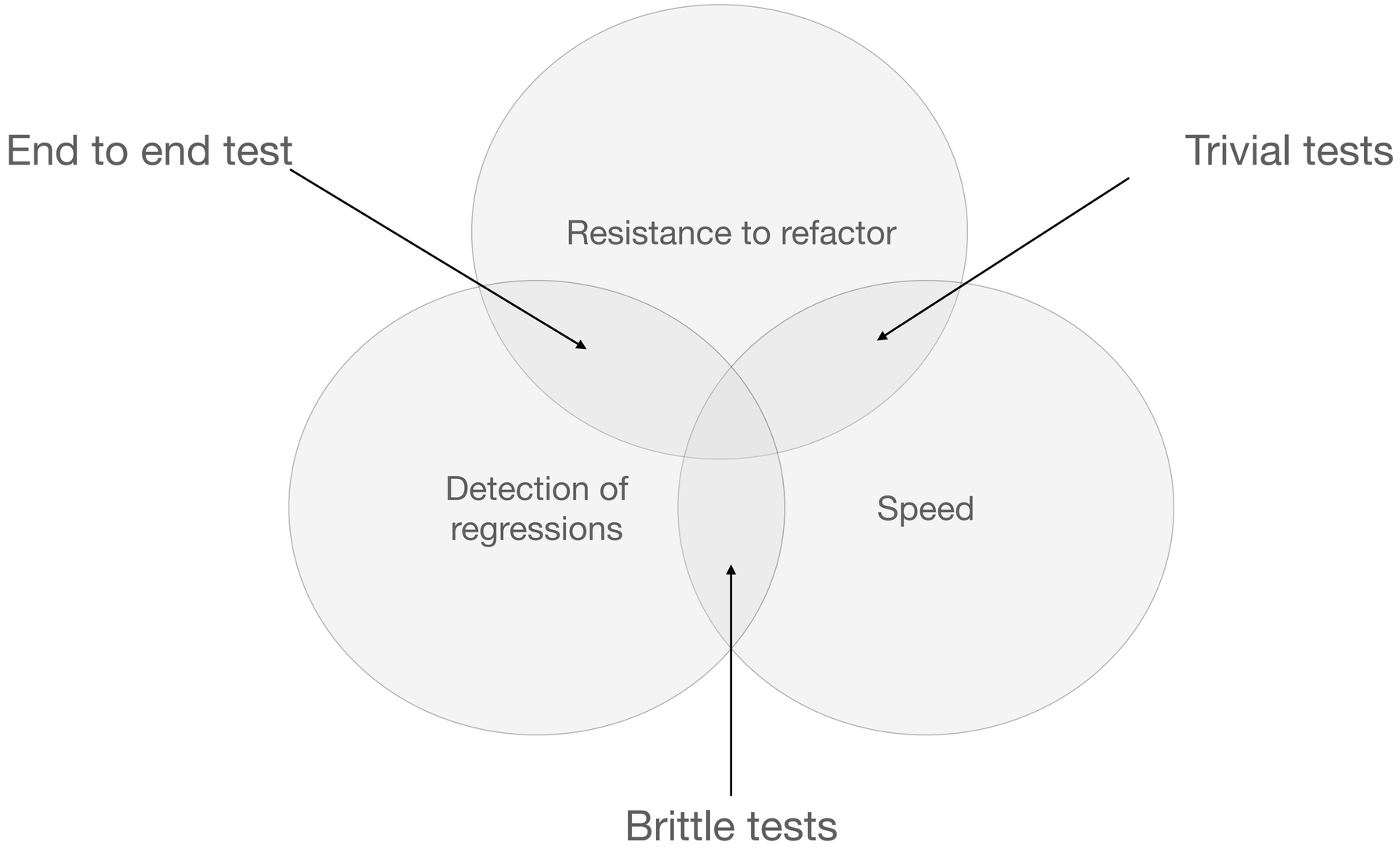
Speed


THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."
MY TESTS ARE RUNNING.
HEY! GET BACK TO WORK!
COMPILING!
OH. CARRY ON.

Resistance to refactor

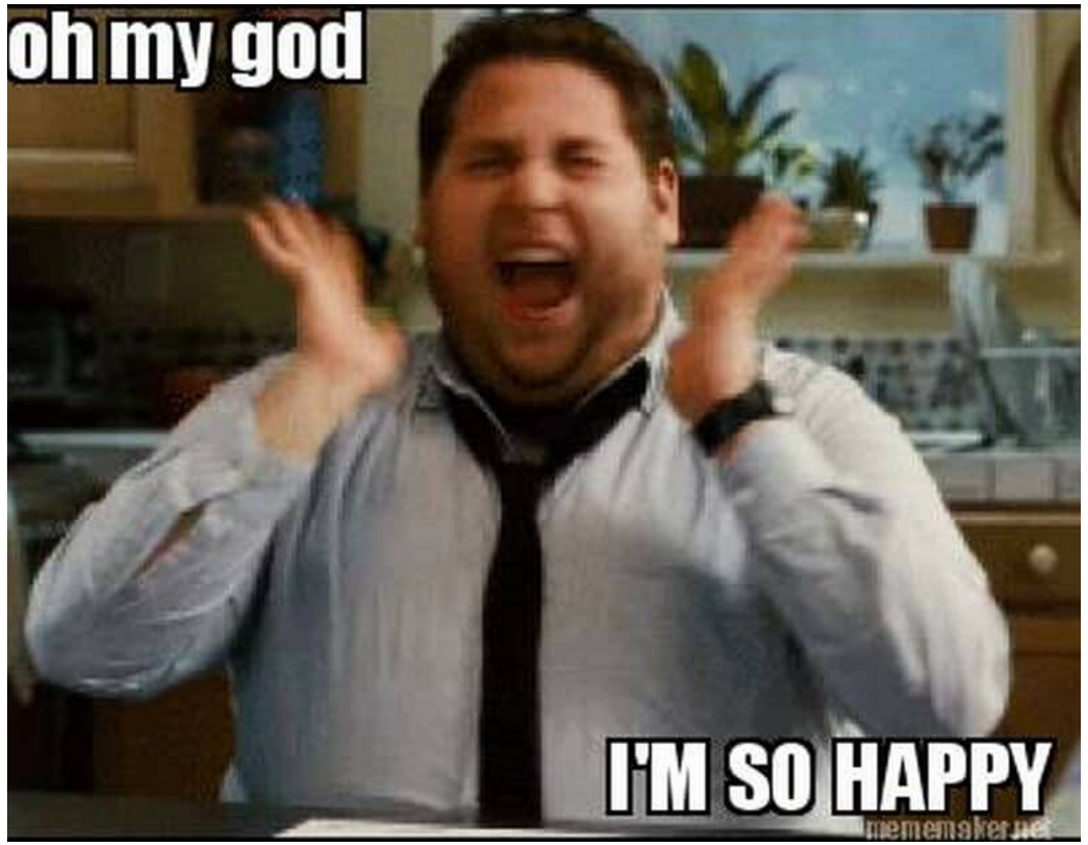
Detection of regression




End to end test
Trivial tests
Resistance to refactor
Detection of regressions
Speed
Brittle tests

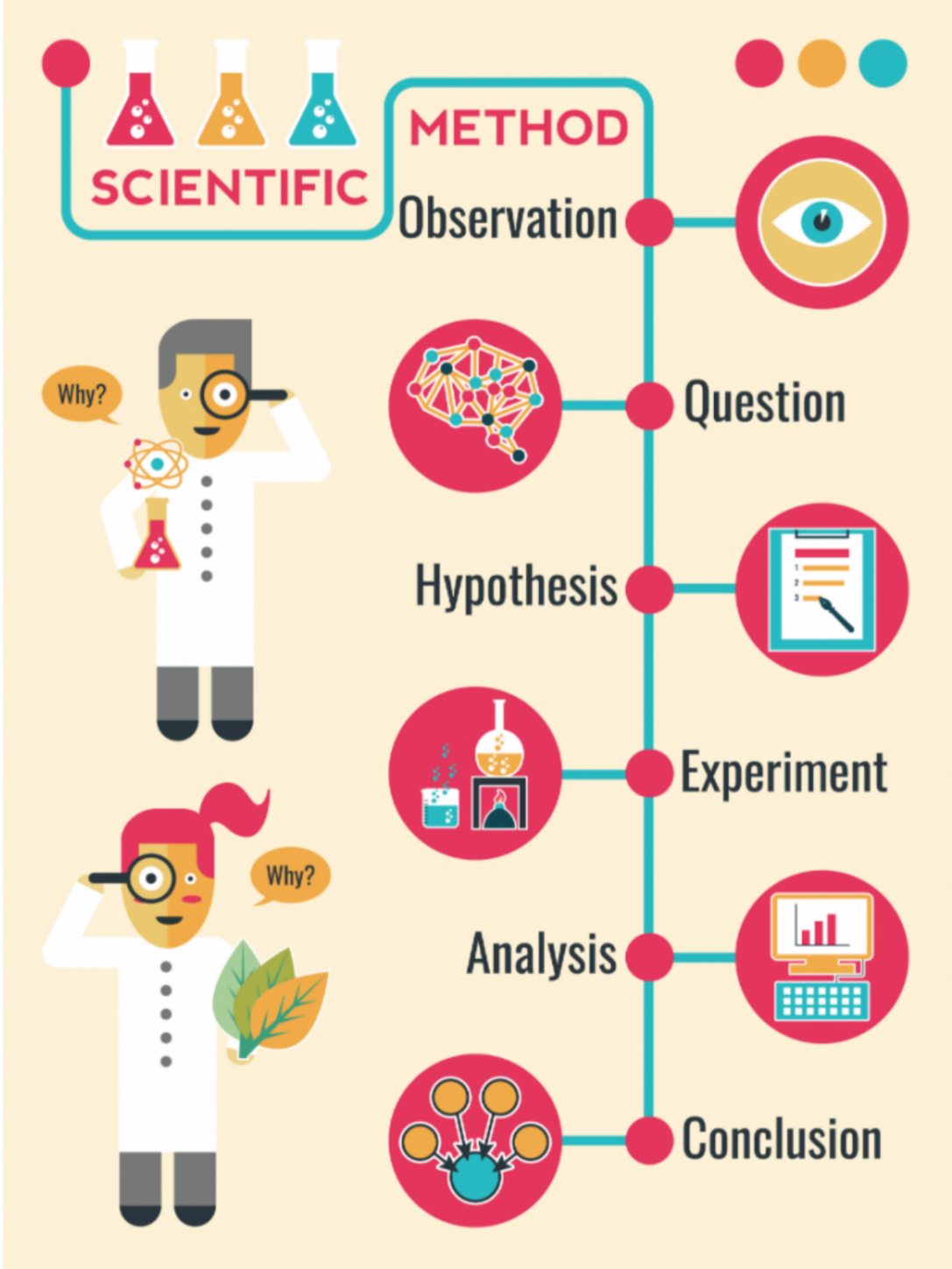
Oh boy my tests are really bad…

I am so happy that my tests are so good…




OPINIONS,
OPINIONS EVERYWHERE
makeameme.

Lets get some data



Speed


THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."
MY TESTS ARE RUNNING.
HEY! GET BACK TO WORK!
COMPILING!
OH. CARRY ON.

DEMO

Resistance to refactor

The Pragmatic Programmers

Investigator:
Date:
Case #:
Location:

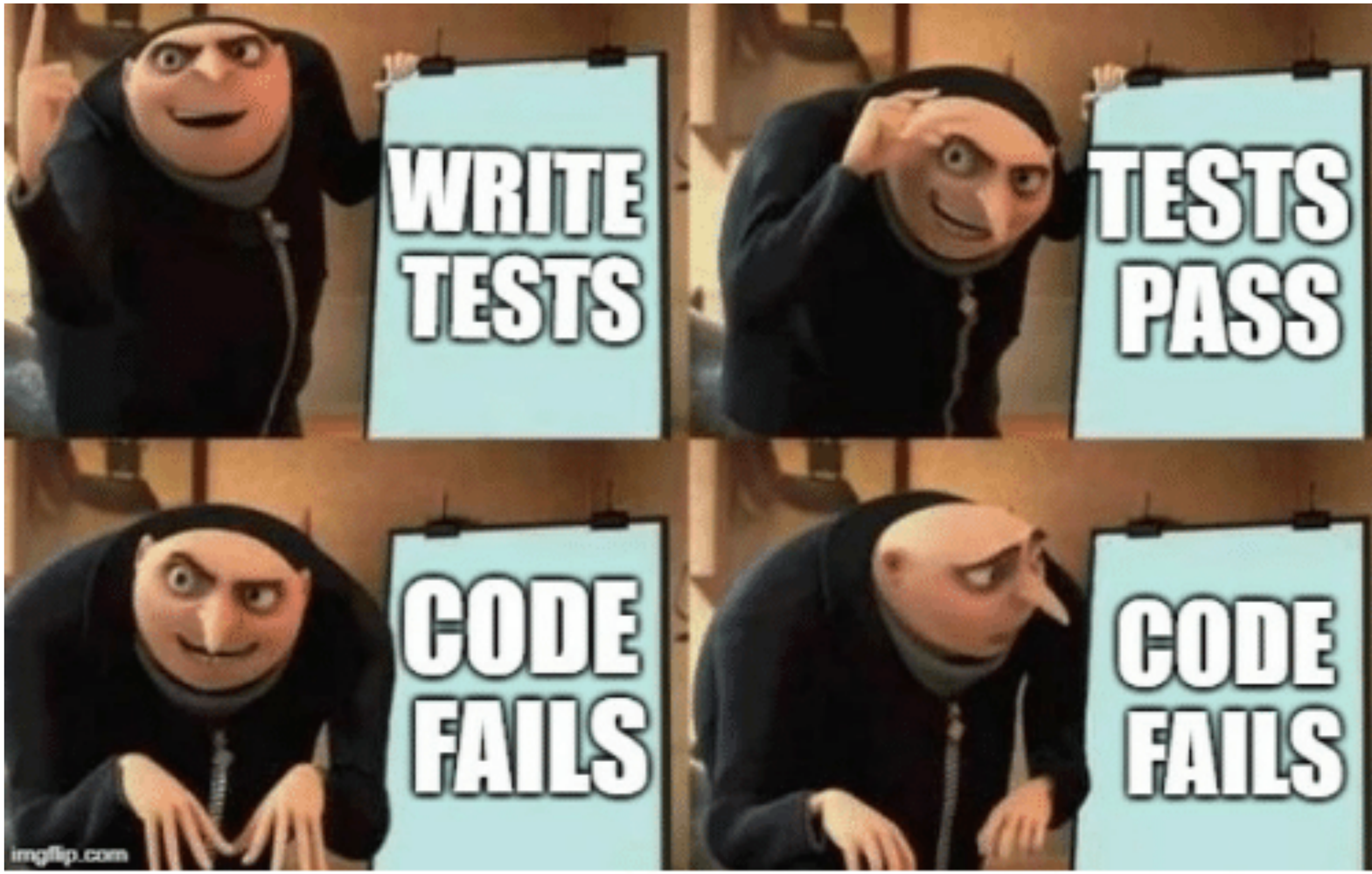
Your Code as a Crime Scene

Use Forensic Techniques
to Arrest Defects, Bottlenecks, and
Bad Design in Your Programs

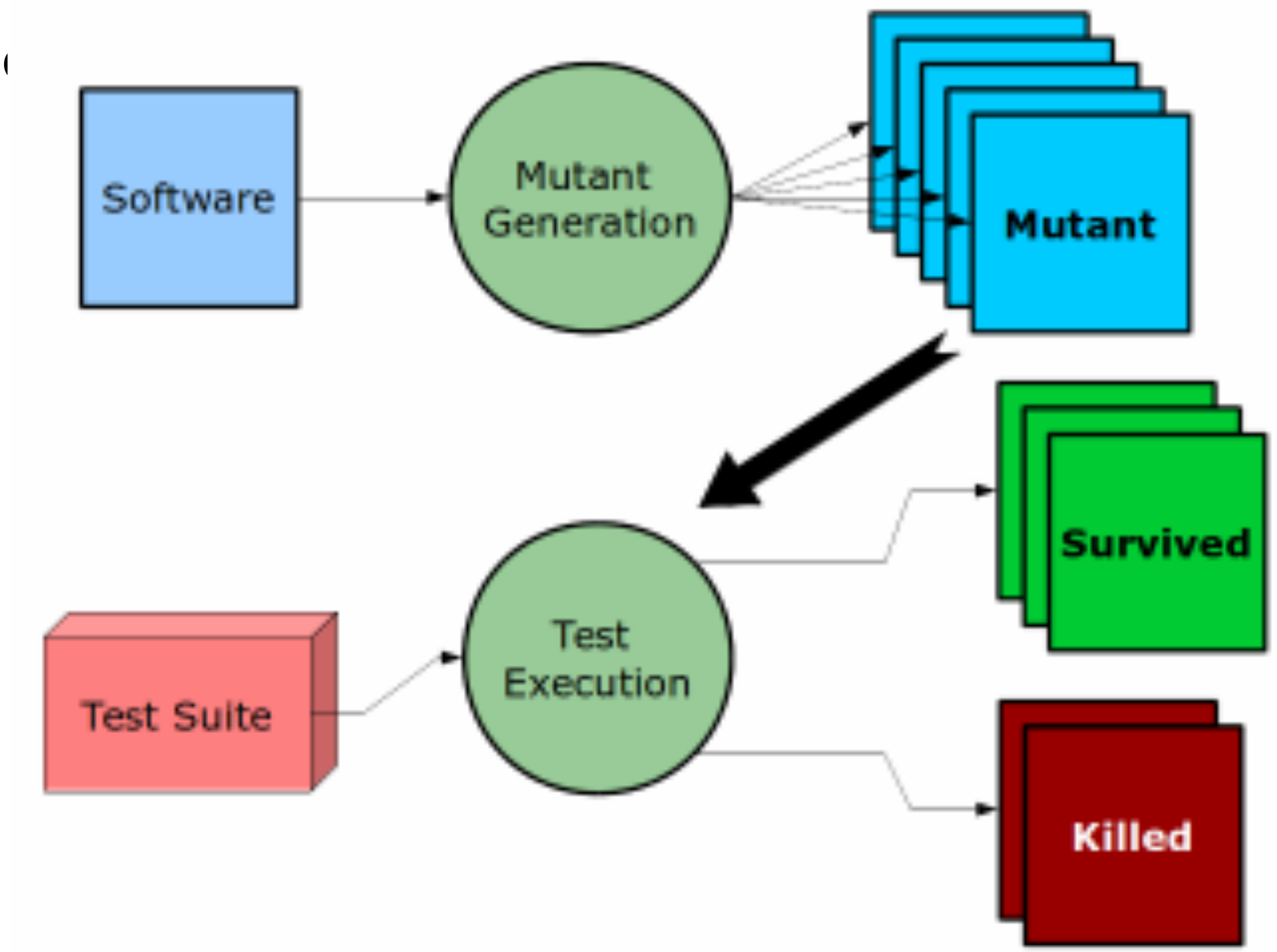
ance to refactor

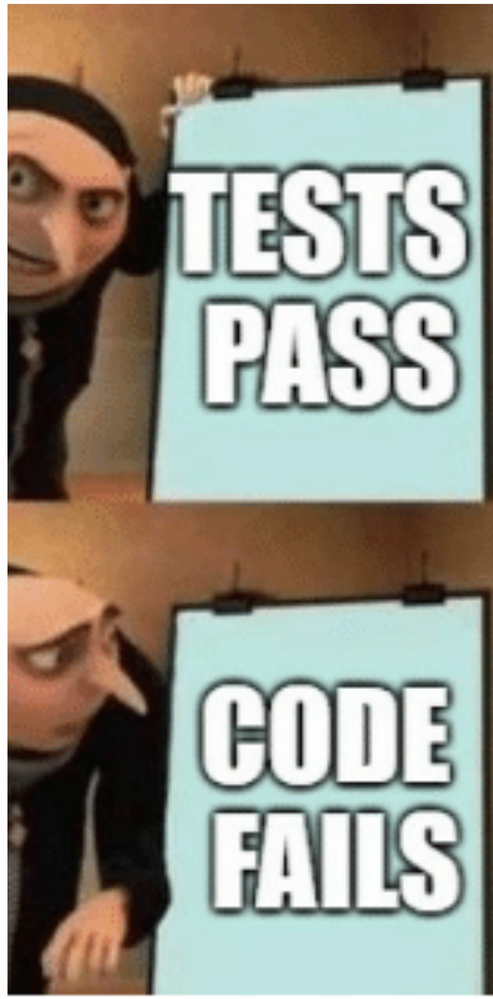
DEMO

Detection of regression



Detection


Software
Mutant Generation
Mutant
Test Suite
Test Execution
Survived
Killed


TESTS PASS
CODE FAILS

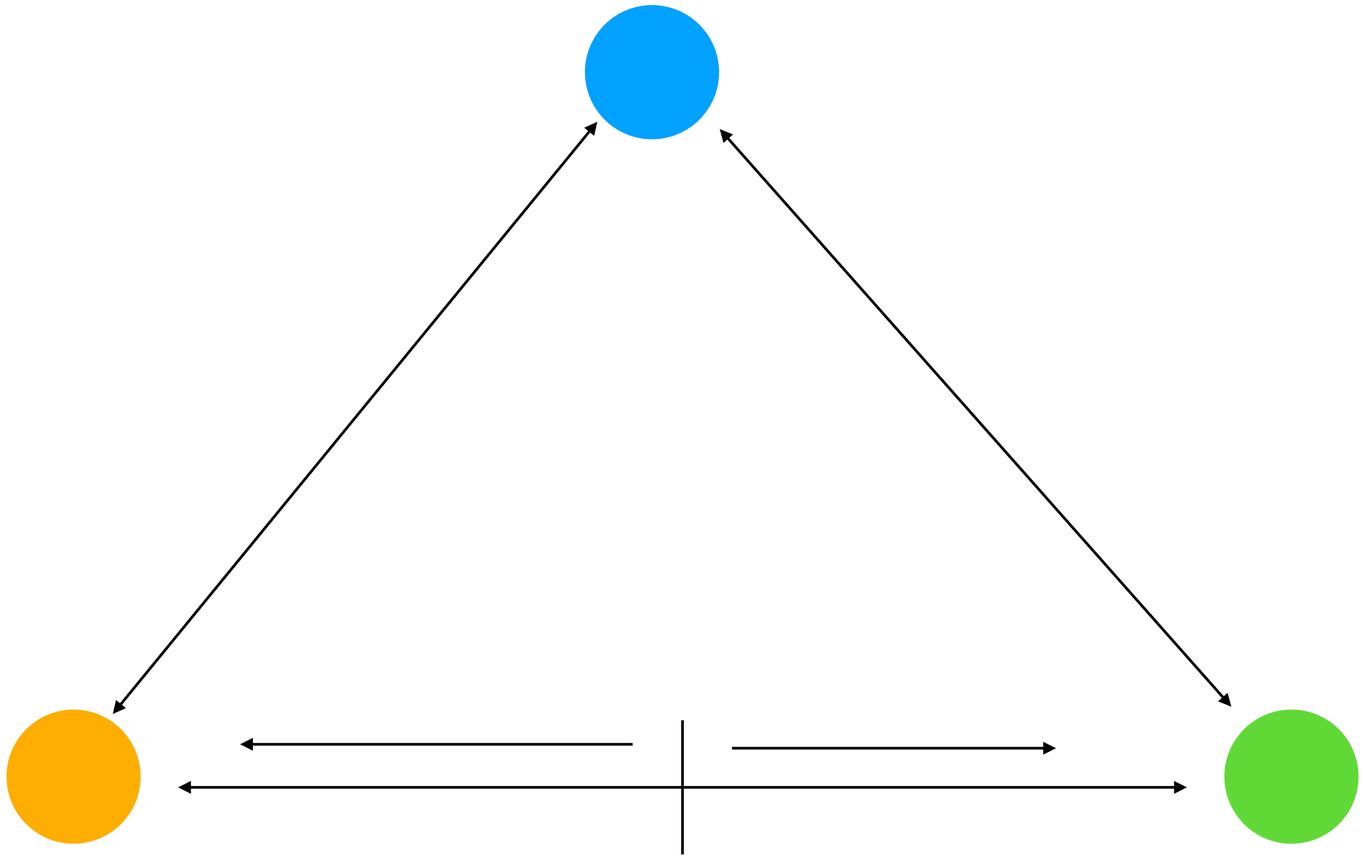
DEMO

I need to start using this right now

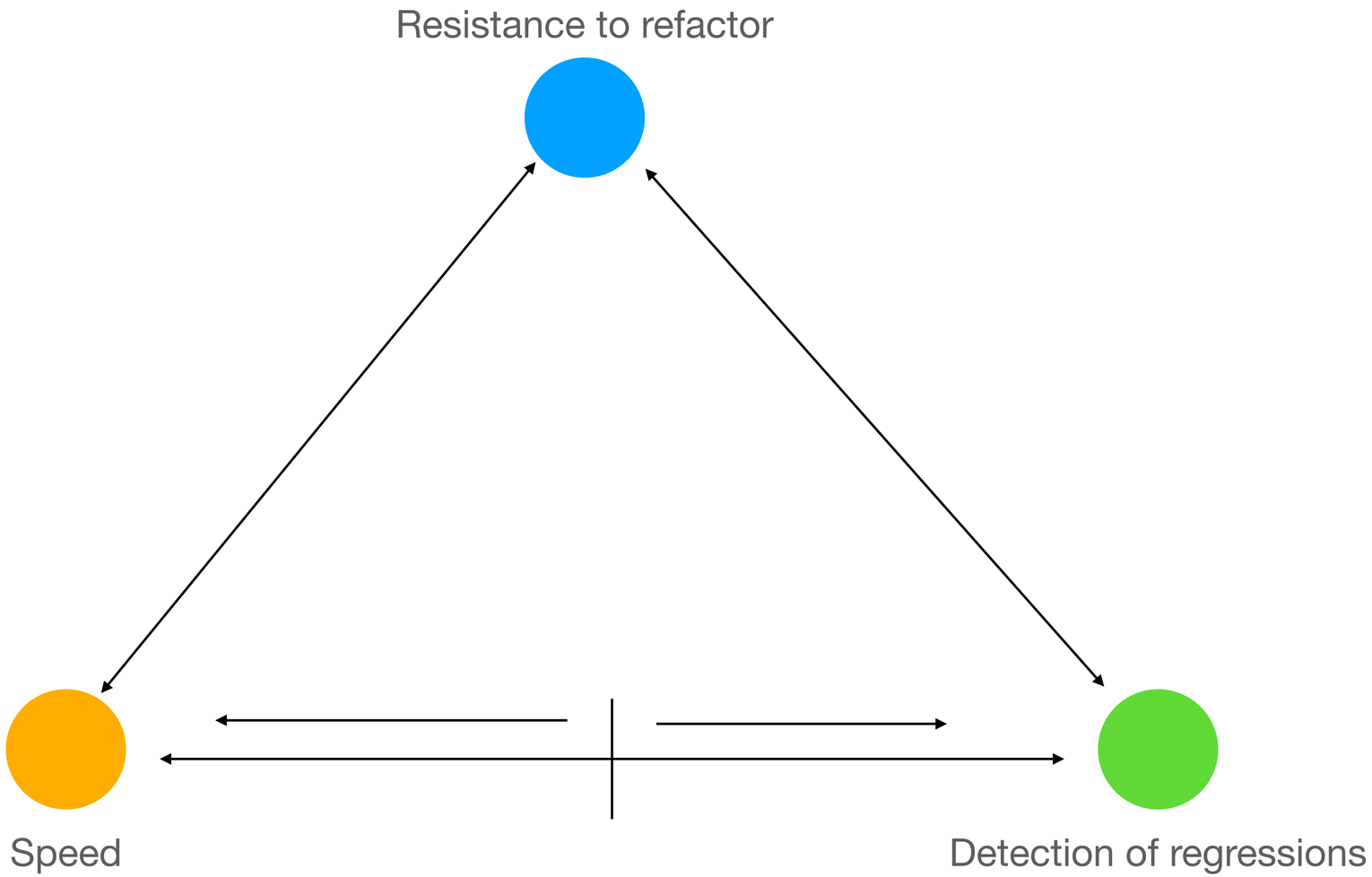
How am I using this data?

To decide how to test something

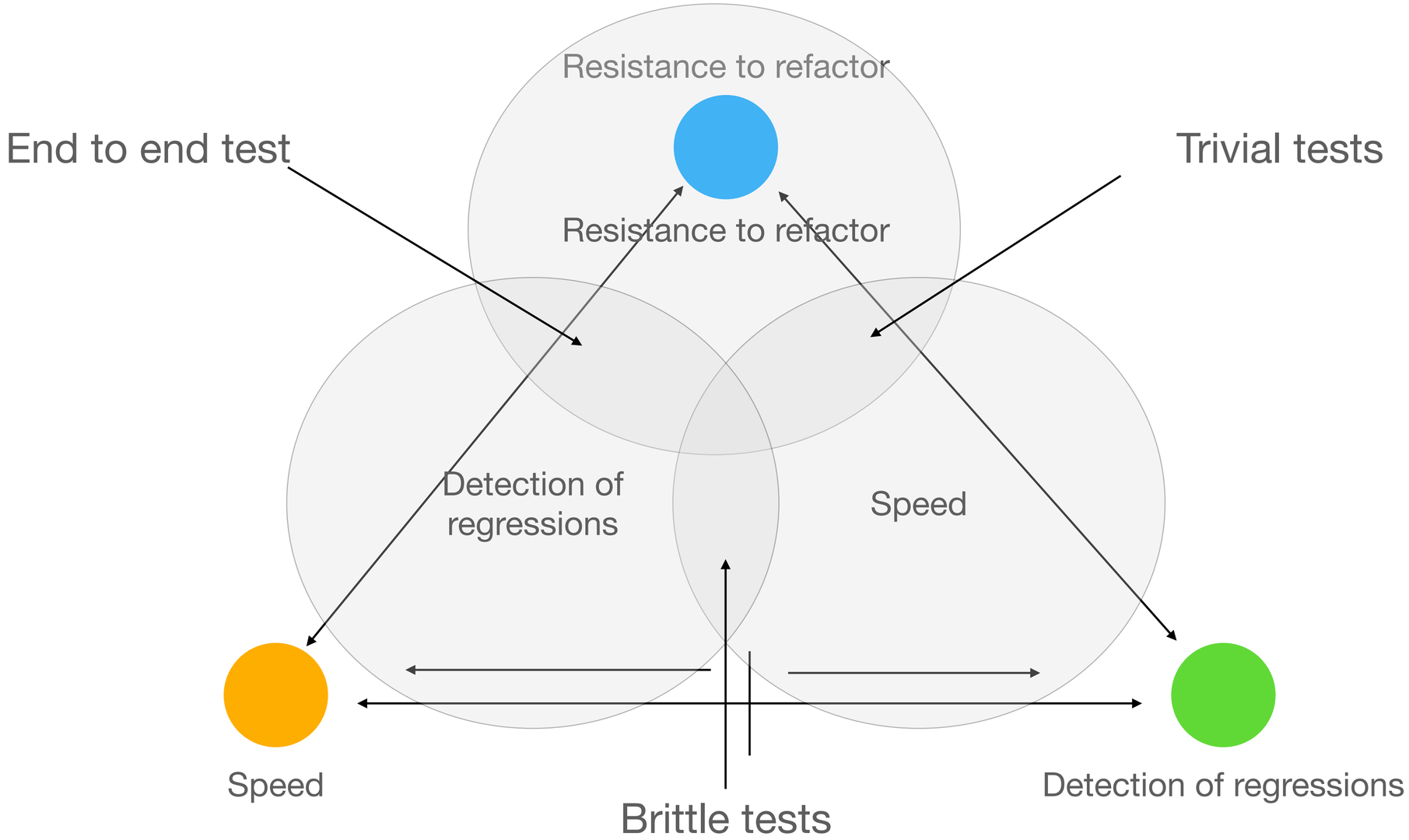
Tune it!



Tune it!

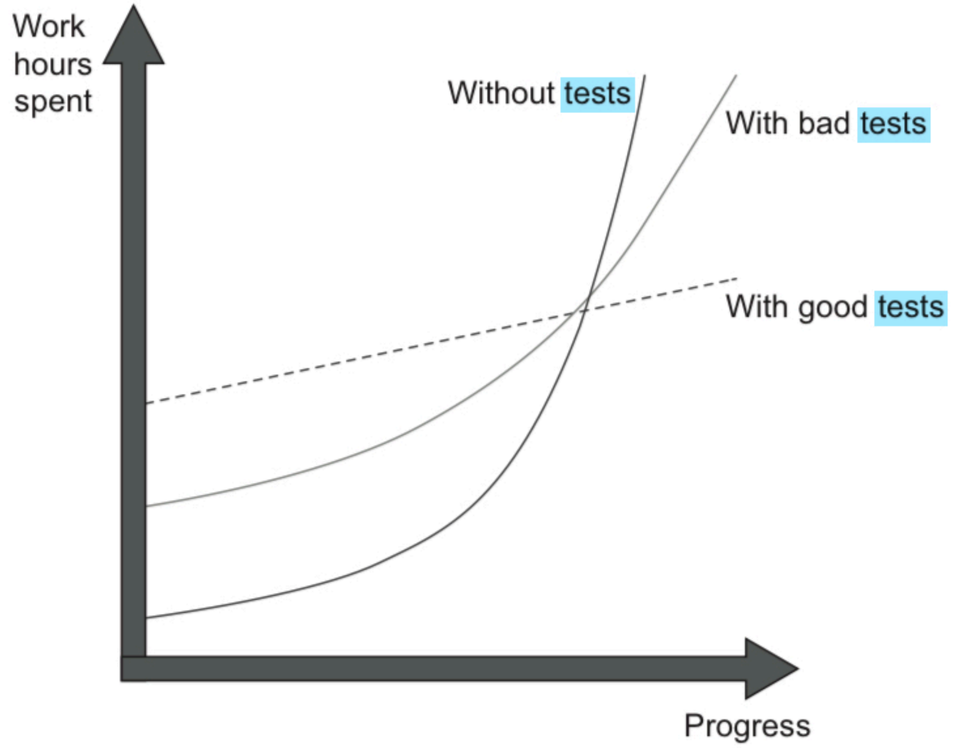


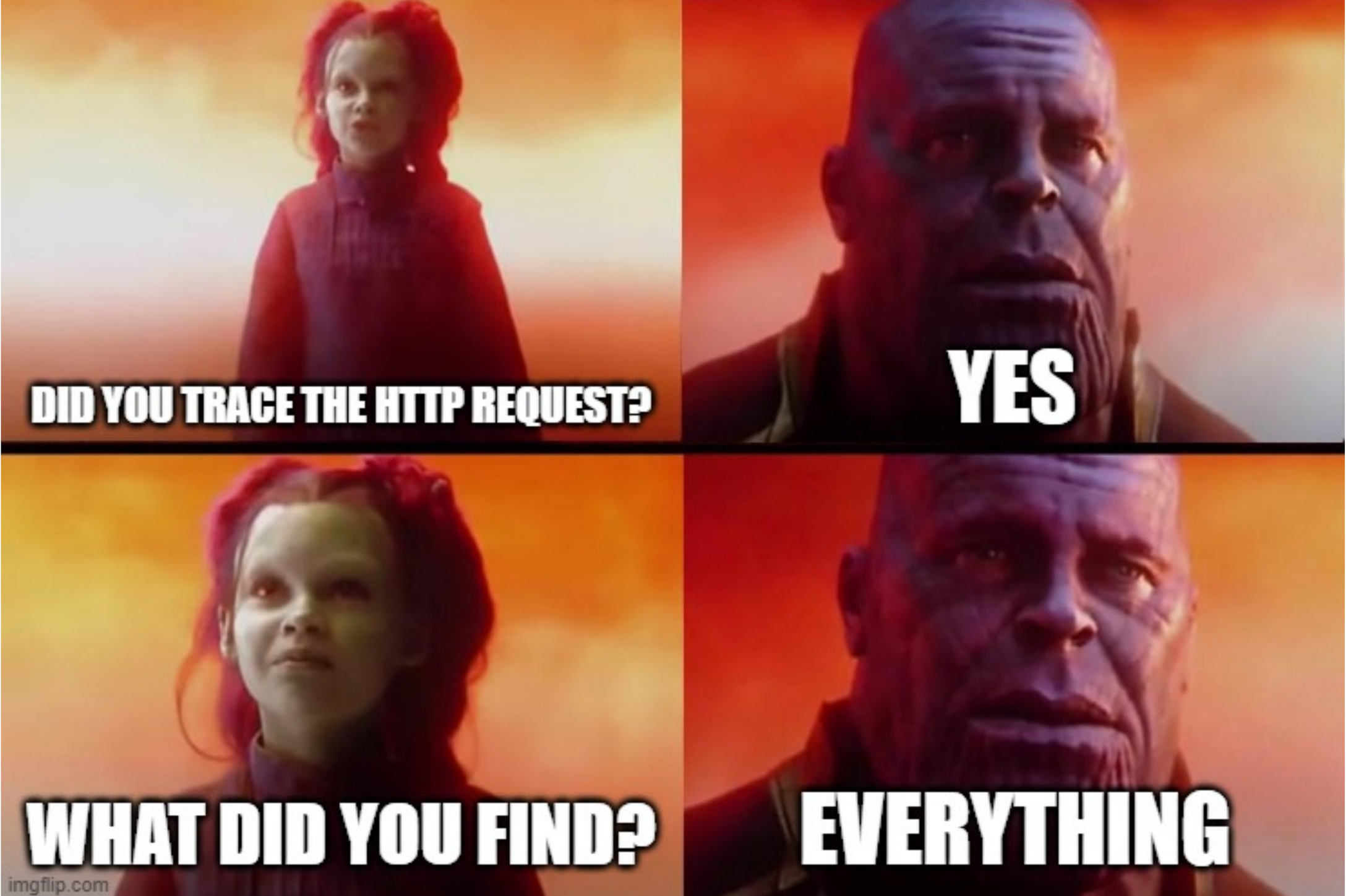
Tune it!


Resistance to refactor
End to end test
Trivial tests
Resistance to refactor
Detection of regressions
Speed
Speed
Brittle tests
Detection of regressions

To decide if write a test is worthy or not

No test is better than a bad test but, then we need something else





Observability

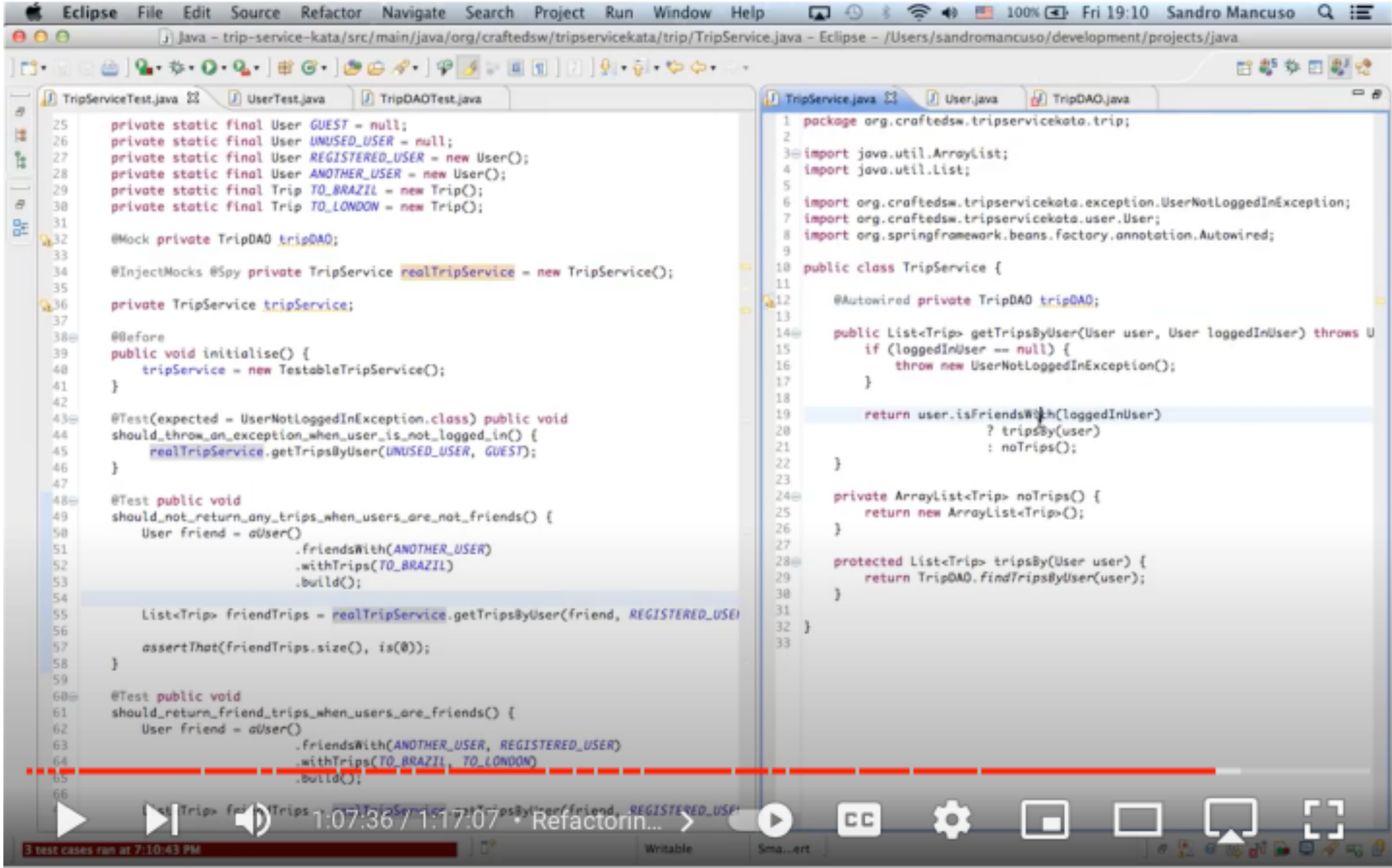
@xuapsdev

@xuaps@fosstodon.org

Do you want an extra ball?

Maintainability

Extra ball


Eclipse File Edit Source Refactor Navigate Search Project Run Window Help
Fri 19:10 Sandro Mancuso
Java – trip-service-kata/src/main/java/org/craftedsw/tripservicekata/trip/TripService.java – Eclipse – /Users/sandromancuso/development/projects/java
TripServiceTest.java
UserTest.java
TripDAOTest.java
TripService.java
User.java
TripDAO.java
private static final User GUEST = null;
private static final User UNUSED_USER = null;
private static final User REGISTERED_USER = new User();
private static final User ANOTHER_USER = new User();
private static final Trip TO_BRAZIL = new Trip();
private static final Trip TO_LONDON = new Trip();
@Mock private TripDAO tripDAO;
@InjectMocks @Spy private TripService realTripService = new TripService();
private TripService tripService;
@Before
public void initialise() {
tripService = new TestableTripService();
}
@Test(expected = UserNotLoggedInException.class) public void
should_throw_an_exception_when_user_is_not_logged_in() {
realTripService.getTripsByUser(UNUSED_USER, GUEST);
}
@Test public void
should_not_return_any_trips_when_users_are_not_friends() {
User friend = aUser()
.friendsWith(ANOTHER_USER)
.withTrips(TO_BRAZIL)
.build();
List<Trip> friendTrips = realTripService.getTripsByUser(friend, REGISTERED_USE
assertThat(friendTrips.size(), is(0));
}
@Test public void
should_return_friend_trips_when_users_are_friends() {
User friend = aUser()
.friendsWith(ANOTHER_USER, REGISTERED_USER)
.withTrips(TO_BRAZIL, TO_LONDON)
.build();
package org.craftedsw.tripservicekata.trip;
import java.util.ArrayList;
import java.util.List;
import org.craftedsw.tripservicekata.exception.UserNotLoggedInException;
import org.craftedsw.tripservicekata.user.User;
import org.springframework.beans.factory.annotation.Autowired;
public class TripService {
@Autowired private TripDAO tripDAO;
public List<Trip> getTripsByUser(User user, User loggedInUser) throws U
if (loggedInUser == null) {
throw new UserNotLoggedInException();
}
return user.isFriendsWith(loggedInUser)
? tripsBy(user)
: noTrips();
}
private ArrayList<Trip> noTrips() {
return new ArrayList<Trip>();
}
protected List<Trip> tripsBy(User user) {
return TripDAO.findTripsByUser(user);
}
}
1:07:36 / 1:17:07 • Refactorin...
CC
3 test cases ran at 7:10:43 PM
Writable

@xuapsdev

@xuaps@fosstodon.org