

TEST IT ALL, TEST IT MORE

COLLABORATION, COMMUNICATION
DEVELOPMENT AND SOME FUN

AGENDA

- Intro of testing tools
- Why Tests
- What type of tests and strategy TDD, BDD
- How to apply those in practice

ANDROID TESTING TOOLS

'DEV' FRAMEWORKS/TOOLS

- JUnit
- testNG
- UI Automator
- Espresso
- Roboelectric

'REMOTE' UI TESTING

- MonkeyRunner
- Monkey Tool
- Robotium
- Appium
- Many non-free solutions...

ASSERTION

- JUnit
- testNG
- FEST
- AssertJ, AssertJ-Android
- Hamcrest
- Truth (Google)

TEST DOUBLES

- Manual
- Using Android's Build Variants
- Mockito
- jMock
- EasyMock
- PowerMock
- Dagger2 ... any DI

WEB SERVER DOUBLES

- WireMock
- MockWebServer
- RESTMock (based on MockWebServer)
- Roboelectric
- mountebank
- Node-RED
- anything else that spits out http responses

'BDD' FRAMEWORKS

- JDave
- Concordion
- Easyb
- JBehave
- Cucumber
- calaba.sh
- SpecFlow
- FitNesse

TYPES OF TESTS

TEST LEVELS

- Unit testing
- Integration testing
- Component interface testing
- System testing
- Operational Acceptance testing

FUNCTIONALITY TESTS

WHY DO WE NEED TESTS ?

NO BUGS?

WHY DO WE NEED TESTS BEFORE DEVELOPMENT ?

FEEDBACK

KNOWING WHAT WE DO

KNOWING WHERE WE GO

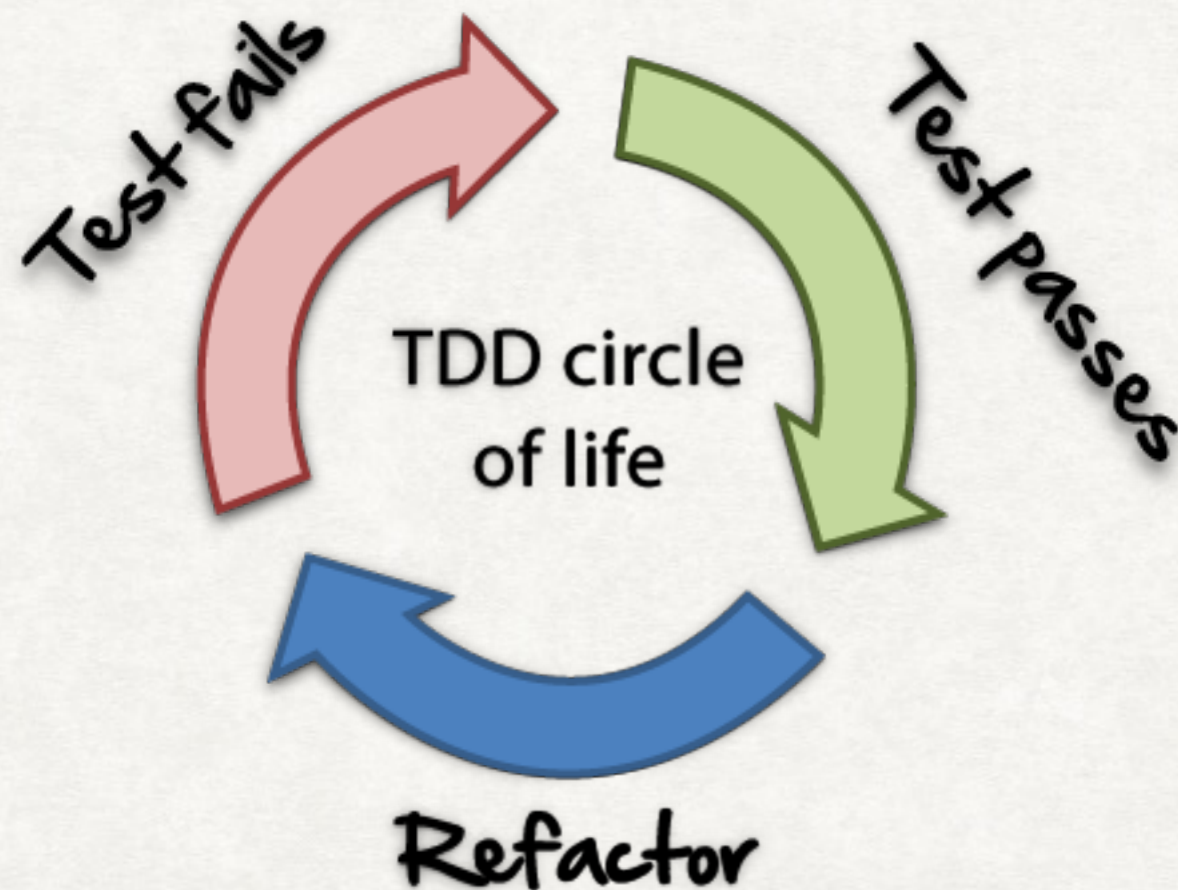
REACT ON TIME

TDD - MYTH OR DEATH

WHAT IS TDD

picture by leantesting.com

- Small iterations
- Wrote/Read by developer
- Fast cycle



IS TDD PERFECT?

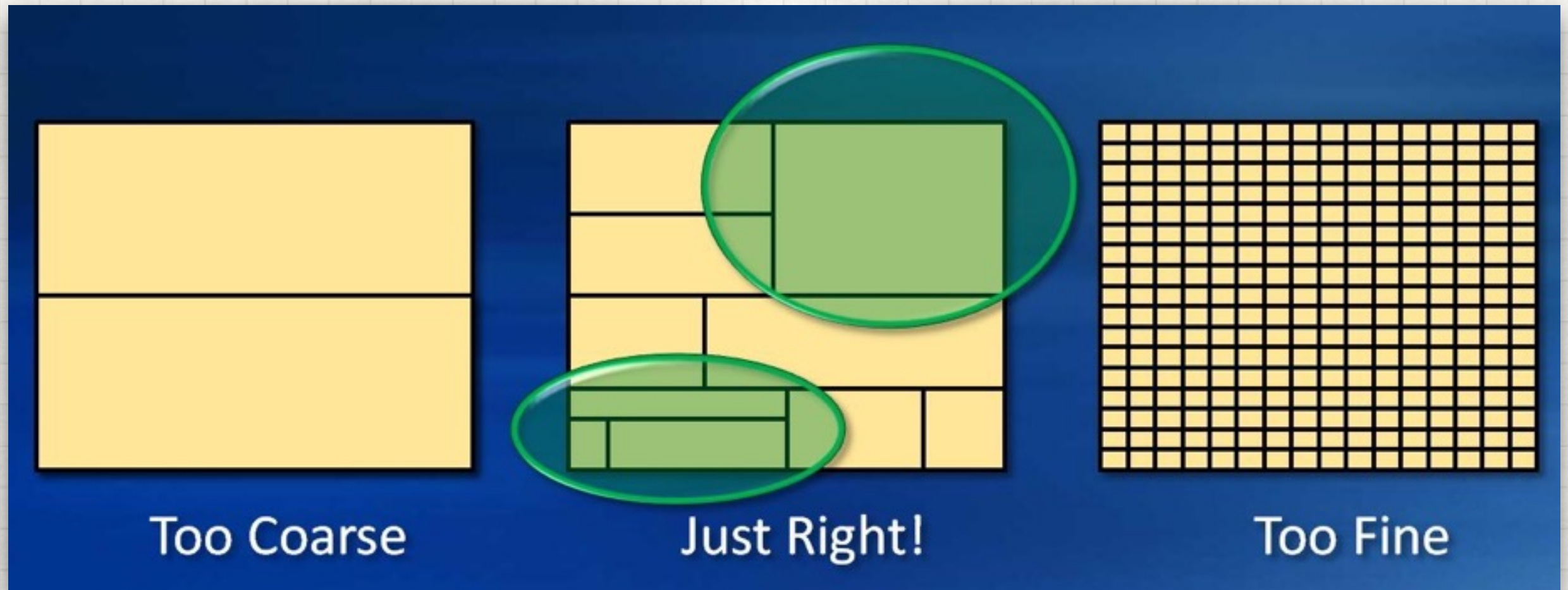


TDD ADDICTION

IS TDD PERFECT?

UNIT
DEFINITION?
....ANYONE?

IS TDD PERFECT?



GRANULARITY

IS TDD PERFECT?

“...UNIT TESTING ALONE CANNOT VERIFY THE FUNCTIONALITY OF A PIECE OF SOFTWARE, BUT RATHER IS USED TO ENSURE THAT THE BUILDING BLOCKS OF THE SOFTWARE WORK INDEPENDENTLY FROM EACH OTHER.

...”

WIKIPEDIA

IS TDD PERFECT?

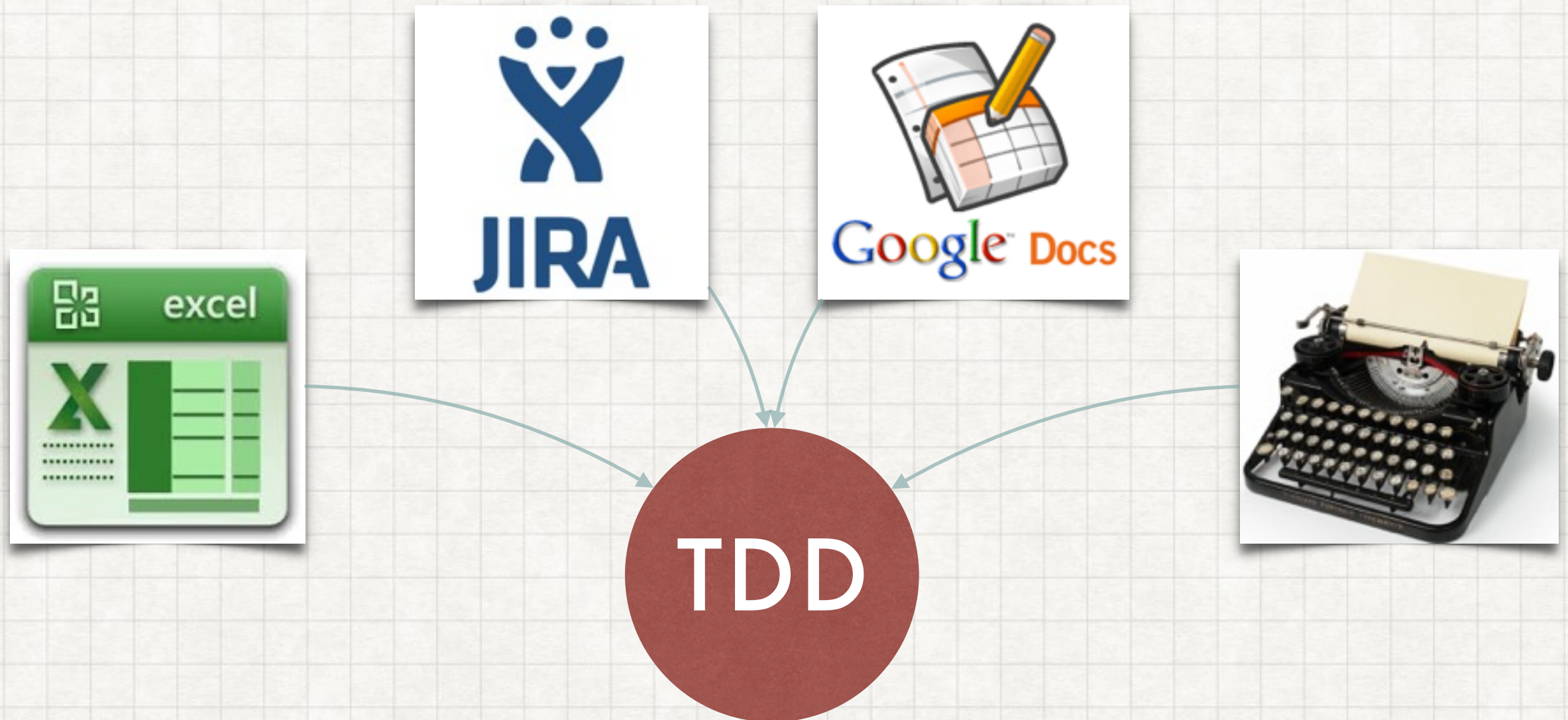


ISOLATION

picture by www.pharmacytimes.com

or lgreensoccer20.wordpress.com

IS TDD PERFECT?



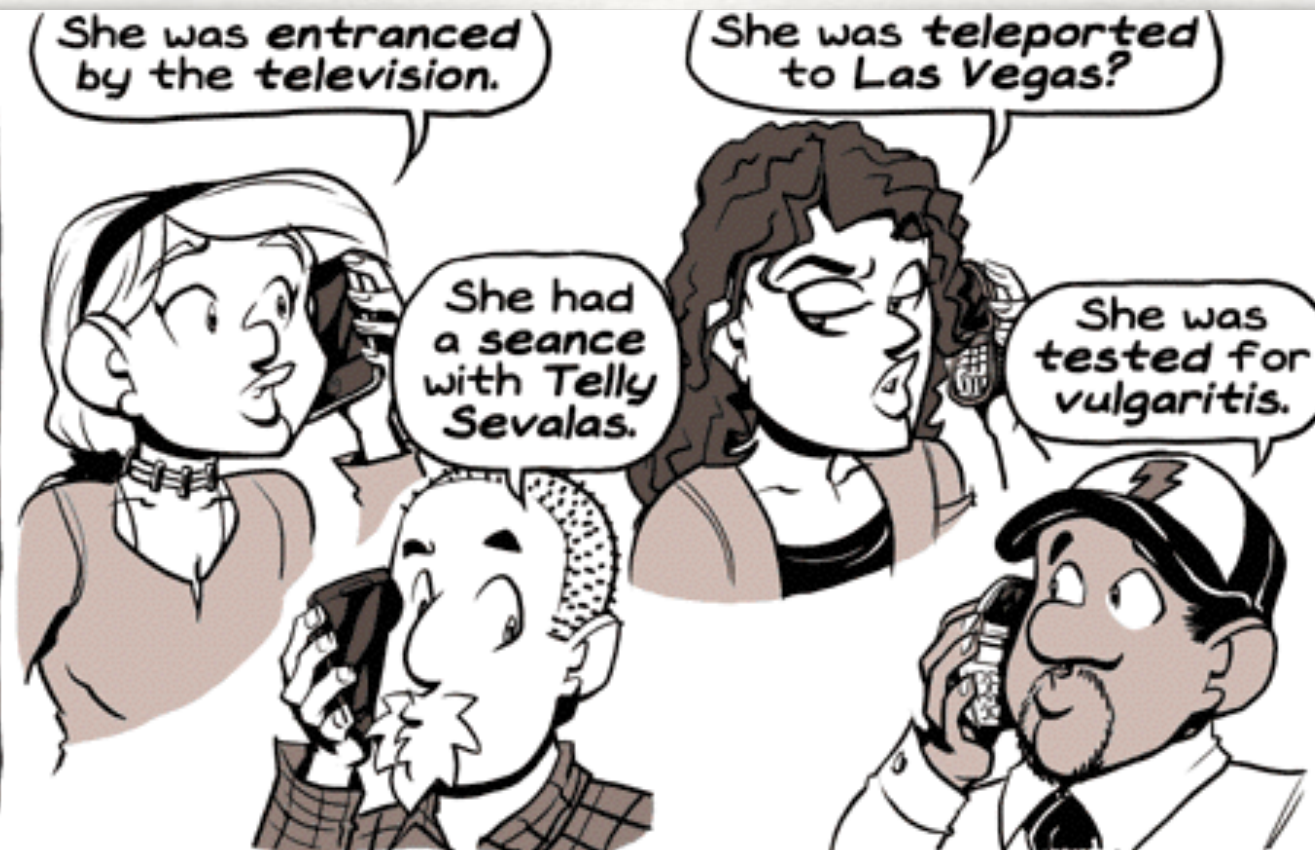
WHERE DO THE SPECS COME FROM?

IS TDD PERFECT?

THE TELEPHONE GAME



www.odoripark.com



©2011 by Chris Watkins

IS TDD PERFECT?

NO, IT ISN'T.

- over-engineering
- no unique unit definition
- granularity
- too much mocks
- specs not centralised
- specs ambiguous

BDD, SBE, ATDD

UNICORNS & FAIRY TAILS?

HOW BDD IS DIFFERENT TDD?

IT'S ALL ABOUT THE
USER

HOW BDD IS DIFFERENT TDD?

IT'S ALL ABOUT
COMMUNICATION

HOW BDD IS DIFFERENT TDD?

IT'S ALL ABOUT
COMMON LANGUAGE

HOW BDD IS DIFFERENT TDD?

IT'S ALL ABOUT
LINKED SPECS

GHERKIN EXAMPLE

Scenario: Some determinable business situation

Given some precondition

And some other precondition

When some action by the actor

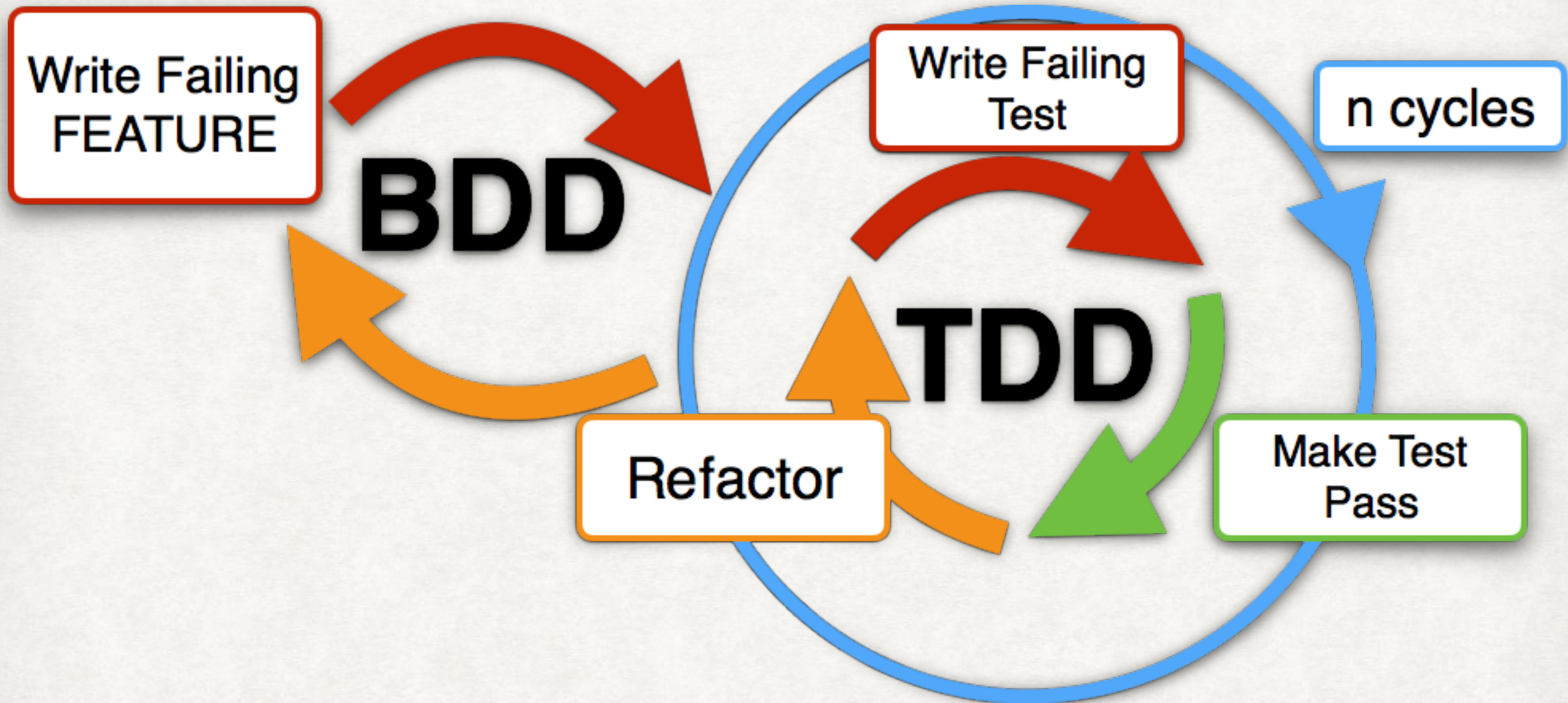
And some other action

And yet another action

Then some testable outcome is achieved

And something else we can check happens too

BDD WORKFLOW



NOW, HOW TO APPLY BDD

IN THE REAL WORLD?

WHAT IS THE BEST WAY?

NO ONE

KNOWS

EXACTLY

WHAT I HAVE LEARNED WITH BDD

PROBLEM

NOT CLEARLY DEFINED TERMS

Given an estate that is sold

When I navigate to property details

Then I should have property sold

SOLUTION

DEFINE THE TERMS WE WORK WITH

Given an estate that is sold

When I navigate to estate details

Then I should have sold field

PROBLEM

INCONSISTENT EXPRESSIONS

SOLUTION

DEFINE POSSIBLE
ACTIONS AND
EXPRESSIONS

PROBLEM

PO, DEV AND QA WRITE
SPECS IN A DIFFERENT
WAY

SOLUTION

DO IT
TOGETHER
(GHERKIN)

SO WE END UP USING **UBIQUITOUS LANGUAGE** BASED ON OUR REFINED **DOMAIN MODEL**

Domain experts should object to terms or structures that are awkward or inadequate to convey domain understanding; developers should watch for ambiguity or inconsistency that will trip up design.

-- Eric Evans (DDD)

...

PROBLEM

**BDD FRAMEWORKS DOES NOT CARE
ABOUT GIVEN WHEN, THEN**

Given estate is sold

When we navigate to Estate Details

Then estate is sold

SOLUTION

**CLEARLY DEFINE WHEN WE SPECIFY
INITIAL STATE, EXECUTE AN ACTION,
VERIFY SOMETHING**

Given there is estate that is sold

When user navigates to Estate Details

Then the estate should be sold

PROBLEM

TEXT AND TRANSLATION
VALIDATIONS BECAME BIG

SOLUTION

USE EXTERNAL FILES FOR
LABELS AND
TRANSLATIONS

PROBLEM

**WE SHARE A REPO OF SPECS
USED BY DIFFERENT DEPARTMENTS**

SOLUTION

WE USE TAGS

@skipiOS

@skipAndroid

@skipWS

PROBLEM

TESTS ARE MANY
CANNOT RUN THEM ALL

SOLUTION

WE USE TAGS

@newiOS

@newAndroid

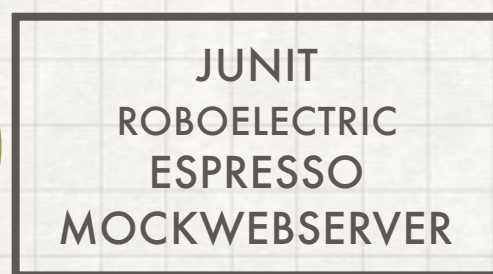
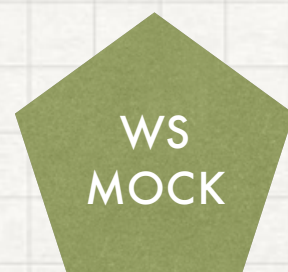
@newWS

PROBLEM

UI TESTS ARE
SLOW

SOLUTION

WE USE
ROBOELECTRIC



**MORE STUFF
TODO**

BETTER GHERKINS GUI EDITOR FOR PO

PARSE AND TEST TRANSLATION FILE

**UNIFY CODE FOR FAST
TDD ITERATION AND REAL
DEVICE TESTS**

WHATS NEXT?

TZATZIKI



<https://github.com/apptik/tzatziki>