Welcome to

TEST DRIVEN DEVELOPMENT For Testers

Workshop code available here:

https://github.com/SarahU/TDDForTestersWorkshop

Who am I?

SARAH

From Johannesburg, South Africa

Developer +/- 7 years

Software Product Development
is a Team Sport

TEST DRIVEN DEVELOPMENT

For Testers

Sarah Usher

@SarahNUsher

#AGILEOTB

@Agileonthebeach

RECIPE

Quick dose of theory

Splash of code

Sprinkling of insights

Rediscovered

Kent Beck

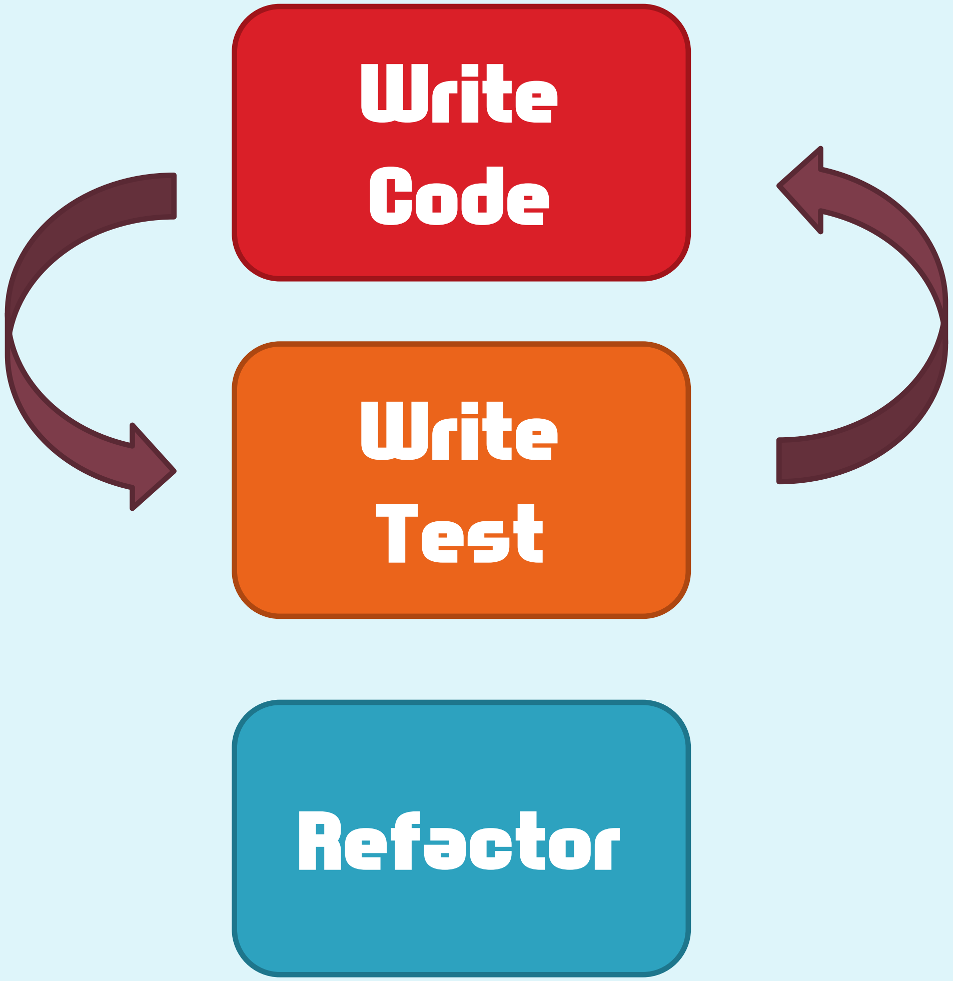
CODE FIRST

Write Code

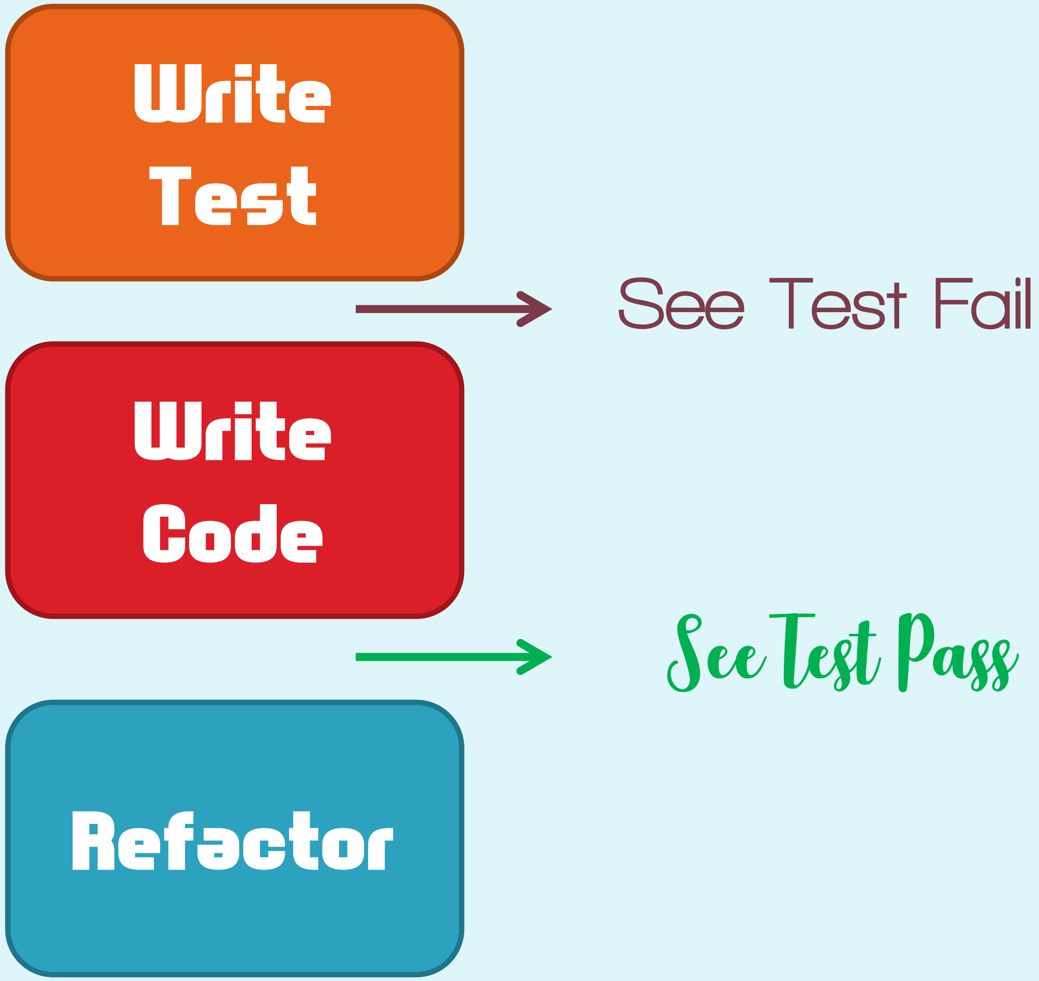
Write Test

Refactor

CODE FIRST



TEST FIRST



So,

what kinds of tests are we talking about here ?

TYPES OF TESTS

- **Unit and Integration Tests**
  Developer TDD

- **Acceptance Tests**
  Acceptance TDD

TYPES OF TESTS

Unit and Integration Tests

Developer TDD $\longrightarrow$ **Developers**

Acceptance Tests

Acceptance TDD $\longrightarrow$ **Testers / Business Analysts / Product People**

TYPES OF TESTS

- **Unit and Integration Tests**

  Developer TDD

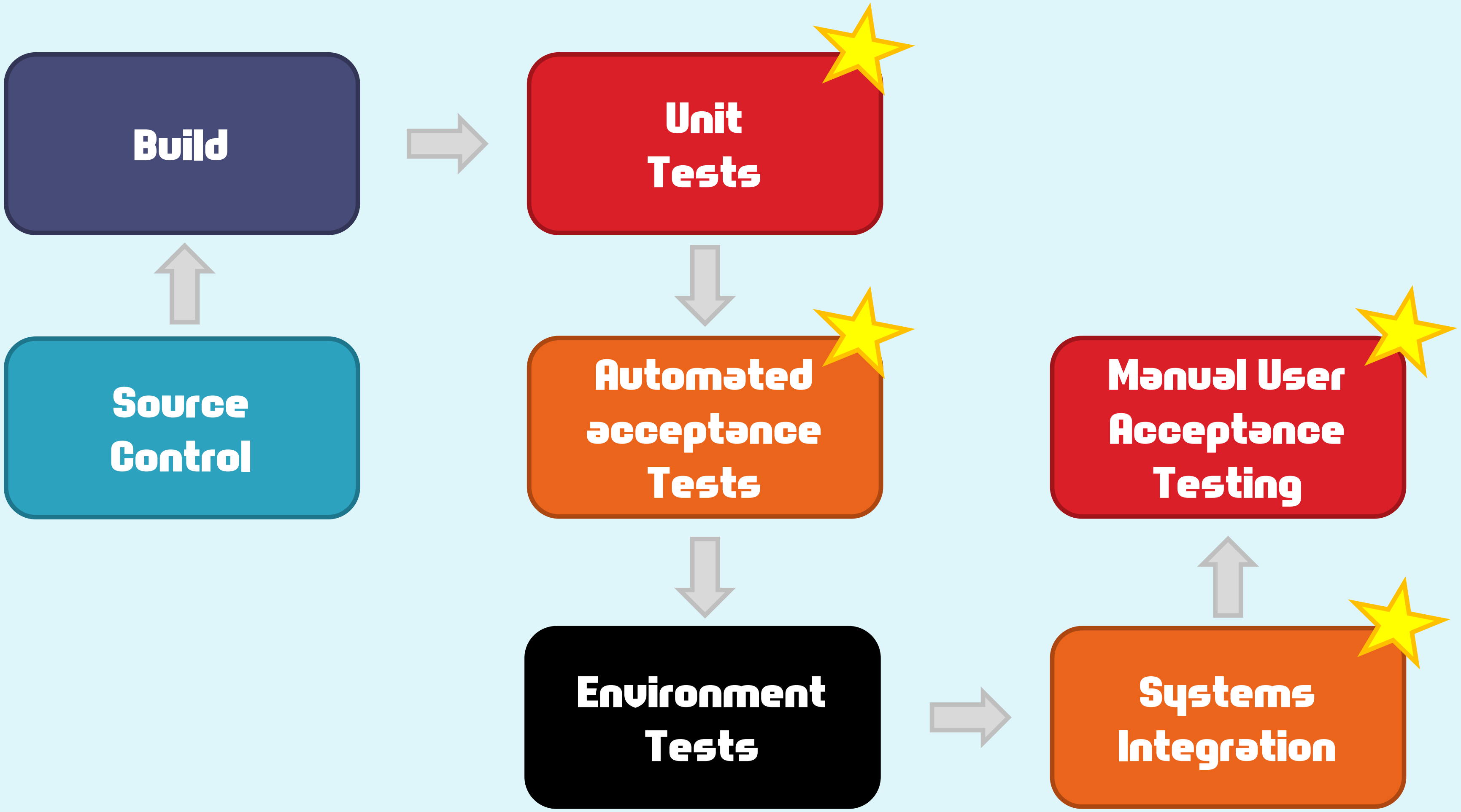
- **Acceptance Tests**

  Acceptance TDD

* **End-to-End Tests – through all layers**

* **Functional & non-functional tests at any layer**

PIPELINE


Build
Unit Tests
Source Control
Automated acceptance Tests
Manual User Acceptance Testing
Environment Tests
Systems Integration

★ Let's see the difference ★

UNIT

VS

INTEGRATION

VS

ACCEPTANCE

Yay, Code!

QA VS DEV USE OF TDD

Devs

Write unit & integration tests to drive the design of code interfaces:

Test Driven *Design*

QAs ?

Developers are very, very good at abstraction. If you give us half a problem, we'll come up with the whole solution. In fact, we're so good at this, we won't even notice that we've only got half the problem.

We're **"solution space" people**. Our job is to solve problems.

Testers, on the other hand, are **"problem space" people**.

They're the ones who ask, "What about X? Or Y? Have you thought about Z?"

They know how to break our code before we've even written it.

If we're really nice to them, they'll tell us.

Their job is to **understand the problem backwards.**

QA VS DEV IN TESTING

QAs

- Upfront acceptance criteria ie. BEFORE any code is written – automated if possible
  (dependant on production code and test automation tooling in use)
- Focused solely on the potential problems / user paths – can determine edge cases, unlikely but possible scenarios, etc…
- Free to be creative and explore the software
- Write tests for issues discovered by users

QA VS DEV IN TESTING

Both (ideally whole team)

- Redundancy is not a bad thing but don't repeat tests unnecessarily – more to clean up later
- Be aware of different tools 😔

QA VS DEV IN TESTING

Both (ideally whole team)

- Redundancy is not a bad thing but don't repeat tests unnecessarily – more to clean up later
- Be aware of different tools 😔

ie:

Put the right tests
in the right places 😄

Set your QA free from manual regression testing

Thank you!

@SarahNUsher

#AGILEOTB

@Agileonthebeach

SOURCES/READING

- https://www.quora.com/Why-does-Kent-Beck-refer-to-the-rediscovery-of-test-driven-development-Whats-the-history-of-test-driven-development-before-Kent-Becks-rediscovery
- https://softwareengineering.stackexchange.com/questions/258311/if-we-have-tdd-and-bdd-why-do-we-need-qa-for/258316