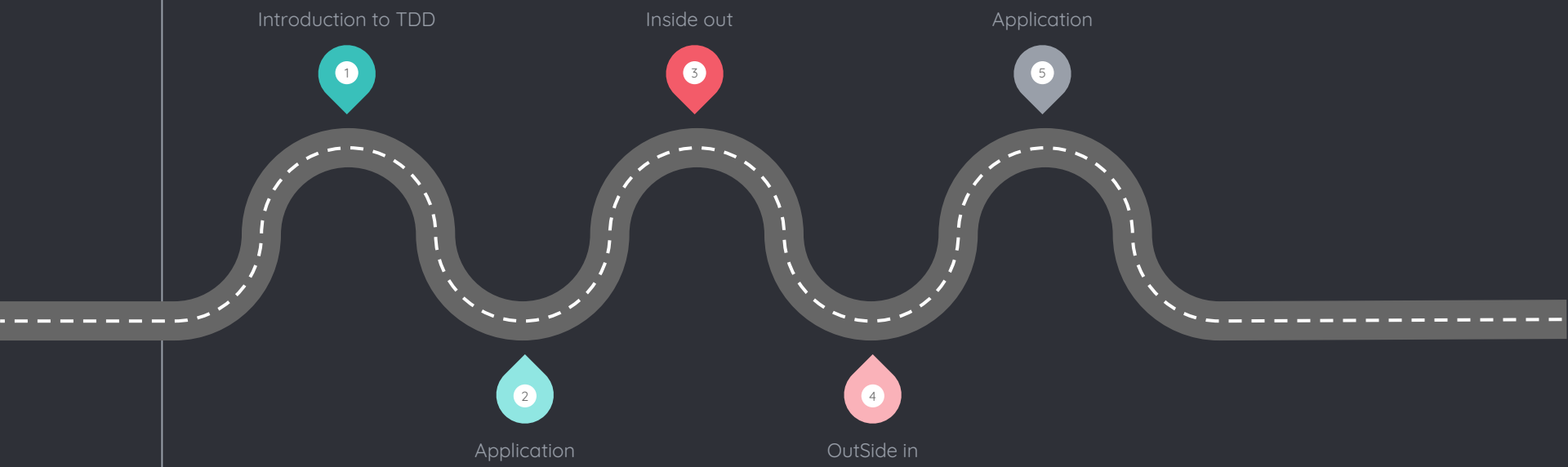
TDD

Test Driven Development

Waleed Elwakeel
Full Stack developer at Orange Egypt

Course Roadmap



1

Introduction to TDD

MSTest

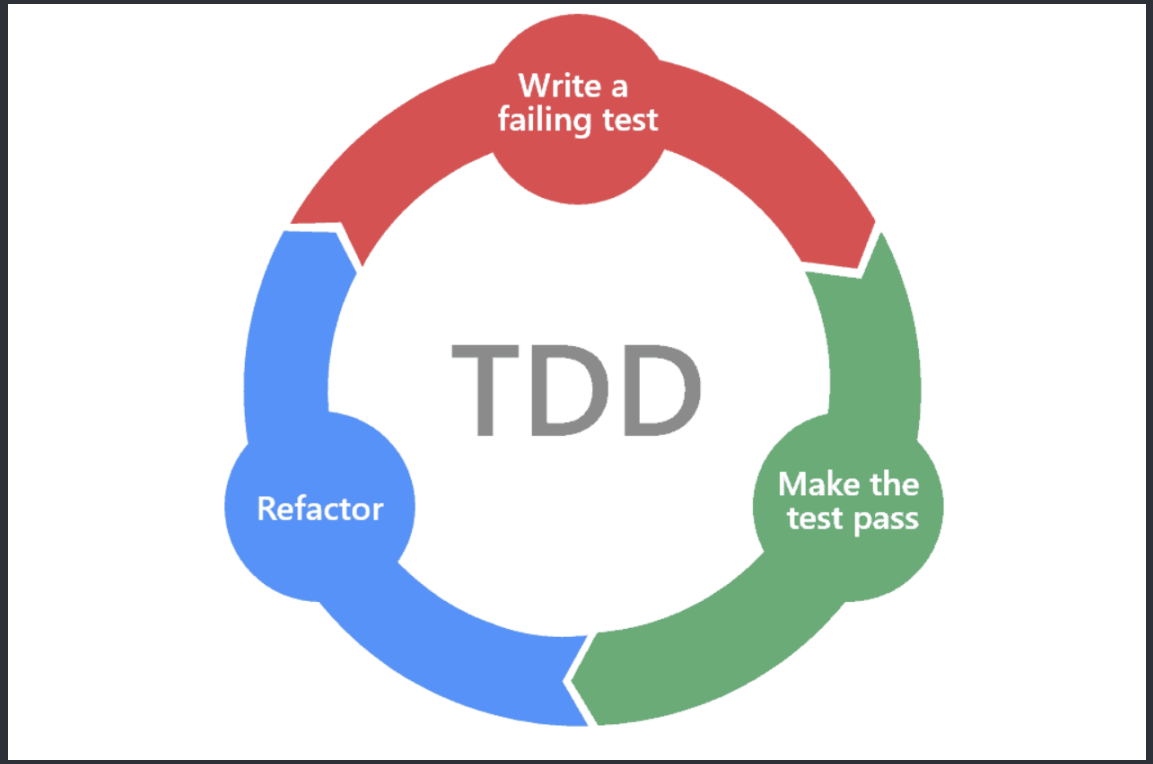
1.1 Common problems:

1. Difficulty in testing code (coverage)
2. Over-engineered solutions
3. Fear of refactoring.

1.2 TDD

Test-driven development (TDD) is a software development process that relies on the repetition of a very short development cycle: first the developer writes an (initially failing) automated test case that defines a desired improvement or new function, then produces the minimum amount of code to pass that test, and finally refactors the new code to acceptable standards.

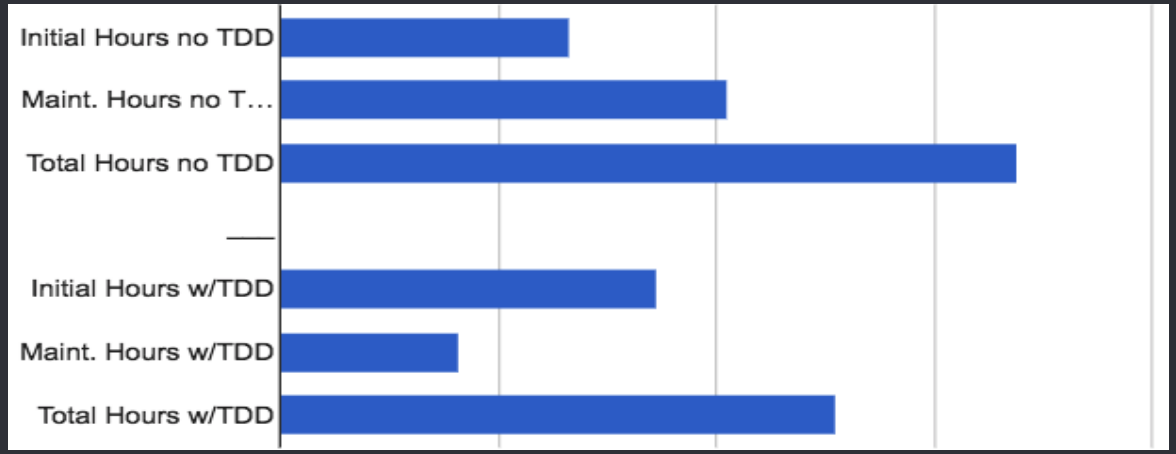
1.3 TDD cycle:



Red - Green - Blue

1.4 Why do we use TDD?

1. Reduce bugs.
2. Avoid code duplication/simple code.
3. Detailed project documentation.
4. TDD reduces the time required for project development
5. No Fear of change.



2 Application

3

Inside out

Inside out

(Bottom Up/Chicago School/Classic Approach)

Inside Out TDD allows the developer to focus on one thing at a time. Each entity (i.e. an individual module or single class) is created until the whole application is built up.

4

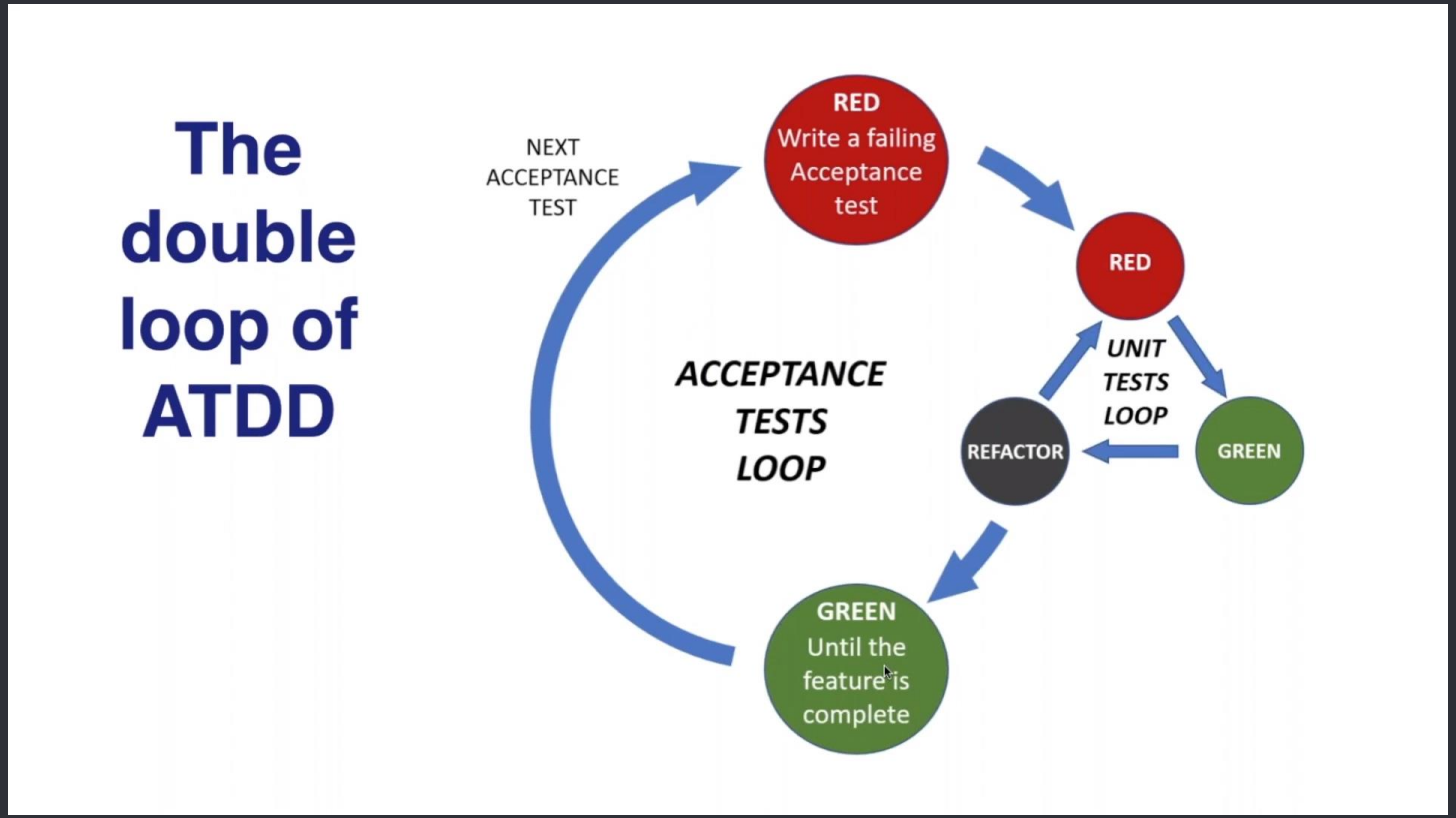
Outside in

4.1 Outside in
(Top Down/London School/Mockist Approach) / ATDD

Outside In TDD depends on having a definable route through the system from the very start, even if some parts are initially hardcoded.

The tests are based upon user-requested scenarios, and entities are wired together from the beginning.

4.2 Outside In Cycle



5 Application

Thanks!

ANY QUESTIONS?

You can find me at

linkedin.com/in/waleedelwakeel

Waleed_elwakeel13@hotmail.me