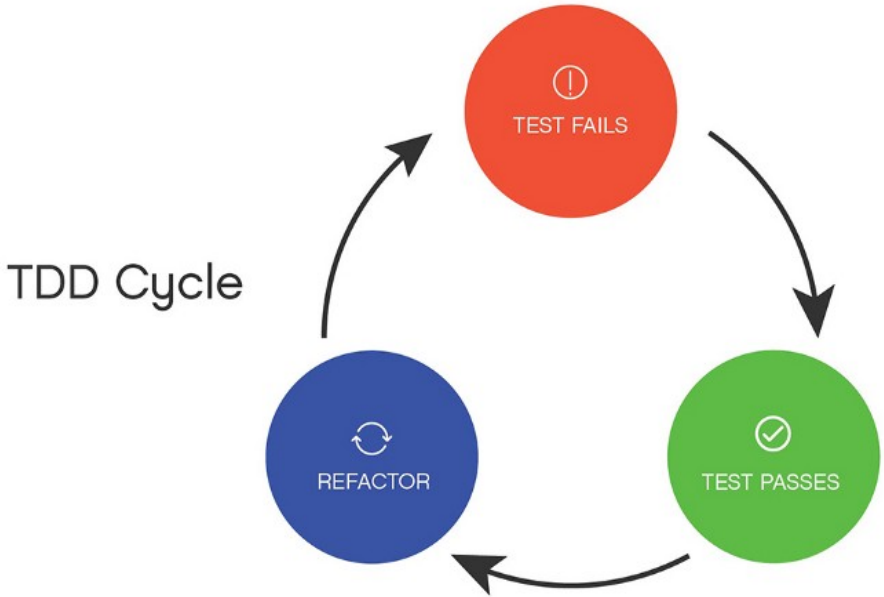
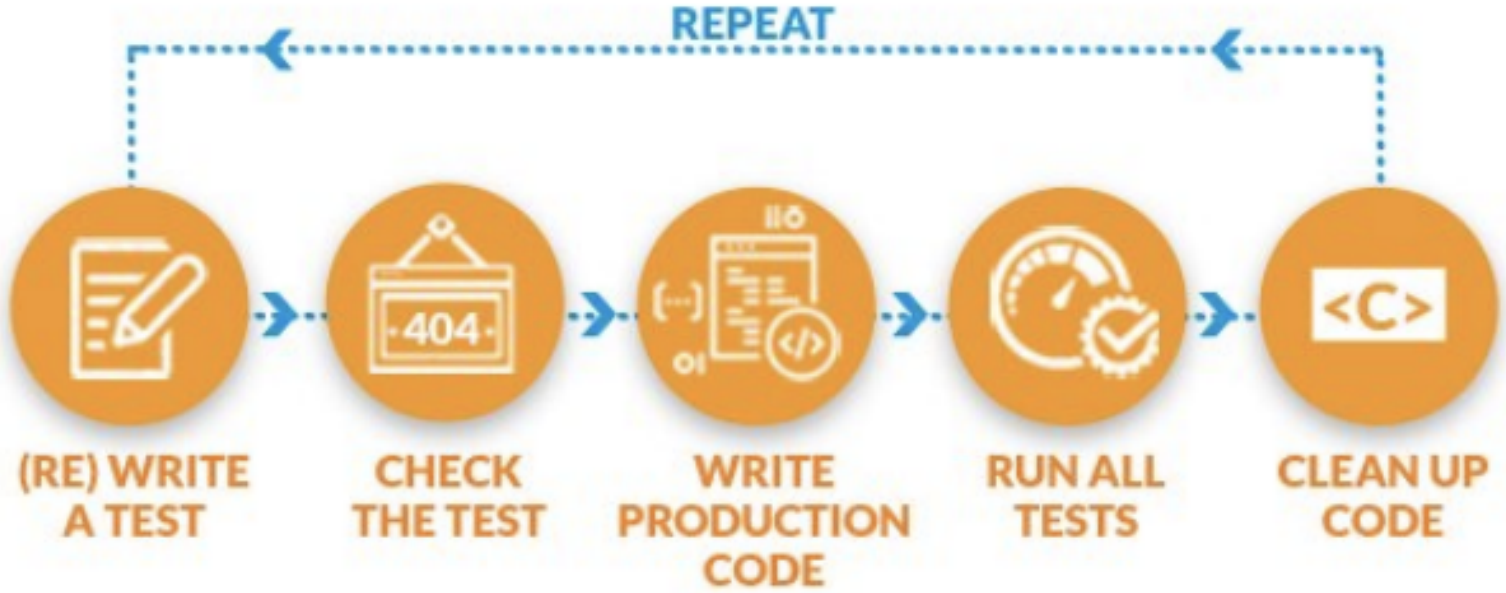
TDD refresher

Slides adapted from originals by Dr Storey, SENG 275

It is driven by **“quality first”** (rather than features first)

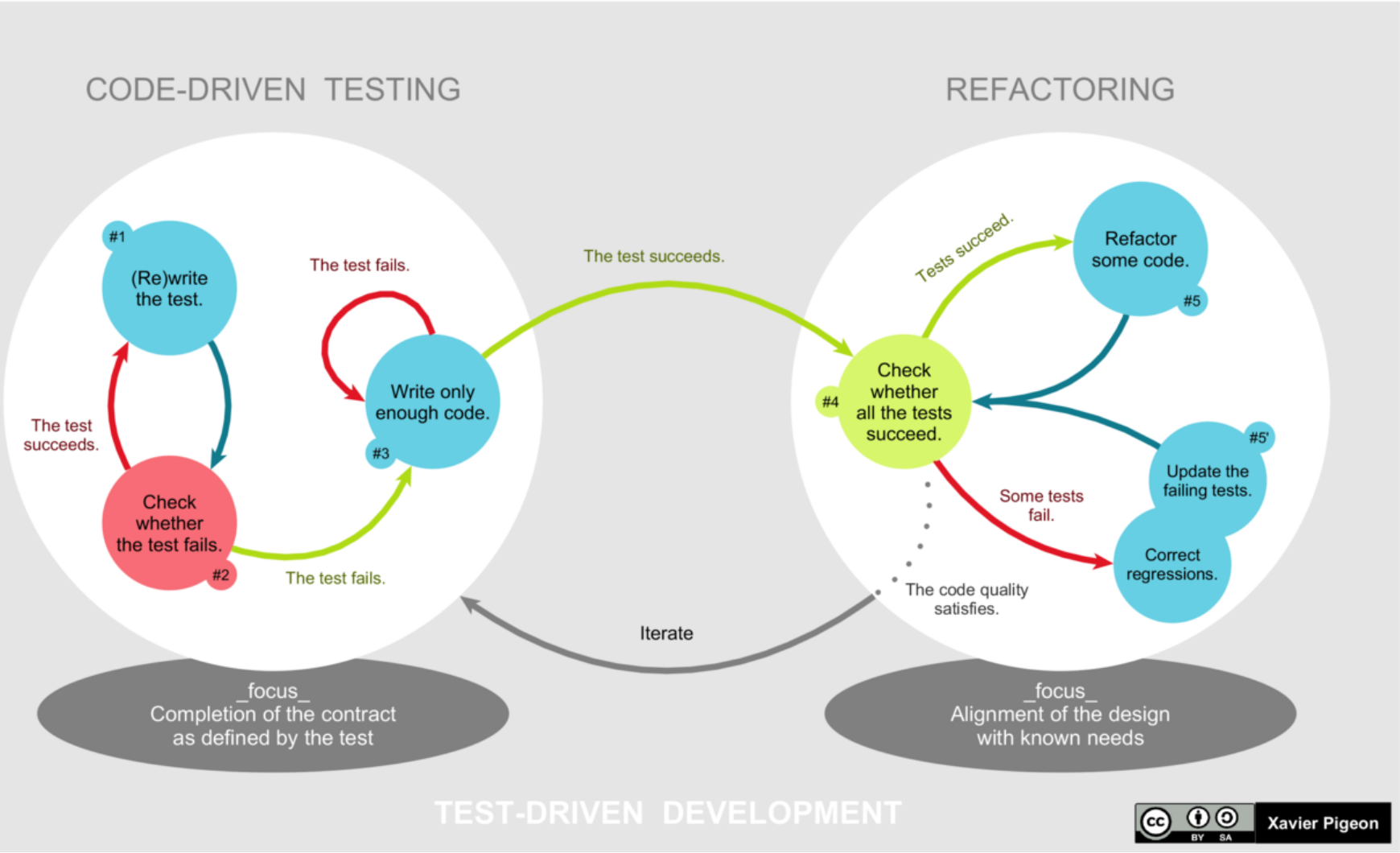
It is about design! TDD is a **three step dance** “red green refactor”





TEST DRIVEN DEVELOPMENT

TDD: not about taking tiny steps, it is about being **able to** take tiny steps (Beck)



Focus on requirements (NEIL: the specification in particular):

makes us think what we expect from a class

how the class should behave in specific cases (if I give it a ‘VII’ what is the rule?)

Reduce useless code

Test from the requirements (not from the code…, but then your code is tested!)

Controls your pace of development (if I feel insecure, I can write a baby step)

We have a first customer!

does the class do what we want, is it easy to use, to set up

Advantages of TDD

Faster feedback

Changing class design is cheaper at the beginning

Forces you to think about managing dependencies from the beginning

“listen to your tests”

Too many tests, perhaps the class is too big?

Too many mocks, perhaps you have too much coupling in your design?

Complex set up, rethink your preconditions!

If tests get too difficult → signal you should refactor!

Advantages of TDD cont.

Note when we do TDD, we are really developing in a different way…
We don’t think about corner cases (boundary values) at the start… we start with the “happy weather tests”

What do ***you*** think about TDD?
Should you do it?

TDD will be a very different approach that requires **discipline** and careful planning.

It works quite well when used with pairing—the navigator can plan out the tests and the driver can focus on the task of passing the tests.

TDD is well suited to backend development because—in most cases—the spec is more well defined than front-end code.

We will have time next week to work on the TDD aspects.

Thinking about the data model and some higher-level design sketching will resolve questions before you have to write tests.

For the Project …

TDD demo (in baby steps… just because we can!):

Let’s look at the Roman Numeral problem - https://github.com/uvic-seng321/tdd-demo