

TDD: Test Driven Development

CPSC 1181 – O.O.

Jeremy Hilliker
Summer 2017

Langara.

THE COLLEGE OF HIGHER LEARNING.

Previously...

- We learned that:
 - The cost to fix issues rises exponentially
- To combat this, we test early (and often)
 - With completely automated regression tests
- Question:
 - Can we do better?
- Idea:
 - What if we test before we write the code?

TDD:

- Write your test before you code.

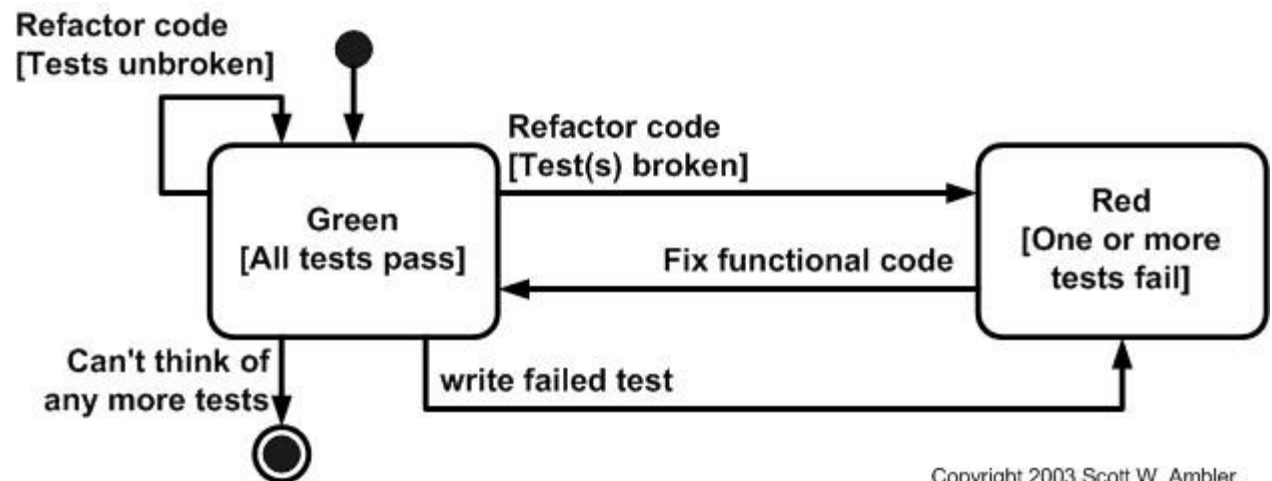
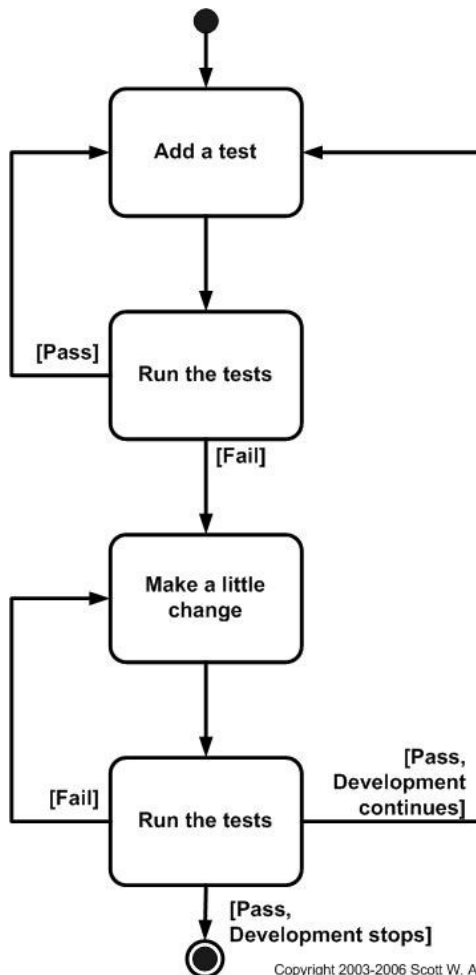
TDD:

- Get a change request
- Make a *small* test
 - No more than needed
 - It should fail (the implementation is not there yet)
- Change the code just enough to pass the test
 - No more!
- Make another small test (that fails)
- Repeat until the change request is fulfilled
- Optional: refactor your code, supported by the tests

Test Driven Development

If it's worth building, it's worth testing.

If it's not worth testing, why are you wasting your time on it?



Why?

- Forces you to think about what this thing is supposed to do
 - No hand-waving allowed
 - If you cant write a test, you don't understand the change
- Forces you to think about how the thing is going to be used
 - You have to use it to write the tests!
- Forces you to design it so that it's easy to test
 - Leads to very high-quality code (low coupling)
- Prevents you from building things you don't need
 - No tests = no additions