The T in TDD
Tests, Types, Tales

Mathias / @brandewinder
Why this talk?

✓ TDD had a deep impact on my coding habits

✓ Since moving to F#, I barely use Red/Green/Refactor
What happened?

✓ Perhaps TDD really is dead?
✓ “doing TDD” differently?
✓ How do I code in F#?
TDD by Example

Kent Beck
Money (classic version)
TODO
if CHF:USD is 2:1,
$5 + 10 \text{ CHF} = \text{$10}
$5 \times 2 = \text{$10}
What’s good about TDD?
Code without fear

_TDD is a way of managing fear during programming._ [...] _If pain is nature’s way of saying “Stop!”, fear is nature’s way of saying “Be careful.” The problem is that fear has a host of other effects:_

✓ _Makes you tentative_

✓ _Makes you grumpy_

✓ _Makes you want to communicate less_

✓ _Makes you shy from feedback_
Small, actual steps

✓ Clear protocol
✓ Break into small chunks
Craft your language

✓ Define vocabulary
✓ Make assumptions explicit
✓ Clarify thinking & conversations
Experiment in code

✓ Write runnable code
✓ Rapid feedback
✓ Surface questions & issues
Feel the pain

✓ Makes bad design painful
✓ Suffer early
Money (F# remix)
The D in TDD

Design, or Development?
Friction

✓ Tests slow down early refactoring
✓ “Premature abstraction”
✓ Sane language defaults
Scripting as TDD

✓ You must design organically, with running code providing feedback between decisions

✓ You must write your own tests, since you can’t wait twenty times a day for someone else to write a test

✓ Your development environment must provide rapid response to small changes

✓ Your designs must consist of many highly cohesive, loosely coupled components, just to make testing easy
A unit test shows one way your code is not broken
A unit test is an anecdote
anecdote
ˈanɪkdəut/
noun
“a short amusing or interesting story about a real incident or person.”
Types

✓ What don’t you test?

✓ Fastest feedback: compile time

✓ Replace a Test with a Type?

✓ Property Based Test: triangulate
Literate Programming

The documentation...

... is the code
Living document

✓ The truth is the code

✓ Working software and comprehensive documentation

✓ F# as a Ubiquitous Language
Conclusion
✓ Happy with classic TDD? Cool.
✓ Do write tests! (pretty please)
✓ Don’t discard ideas wholesale
✓ Do look at other paradigms
I'll give you my gun REPL when you pry it from my cold, dead hands
Thank you!

Mathias Brandewinder
@brandewinder