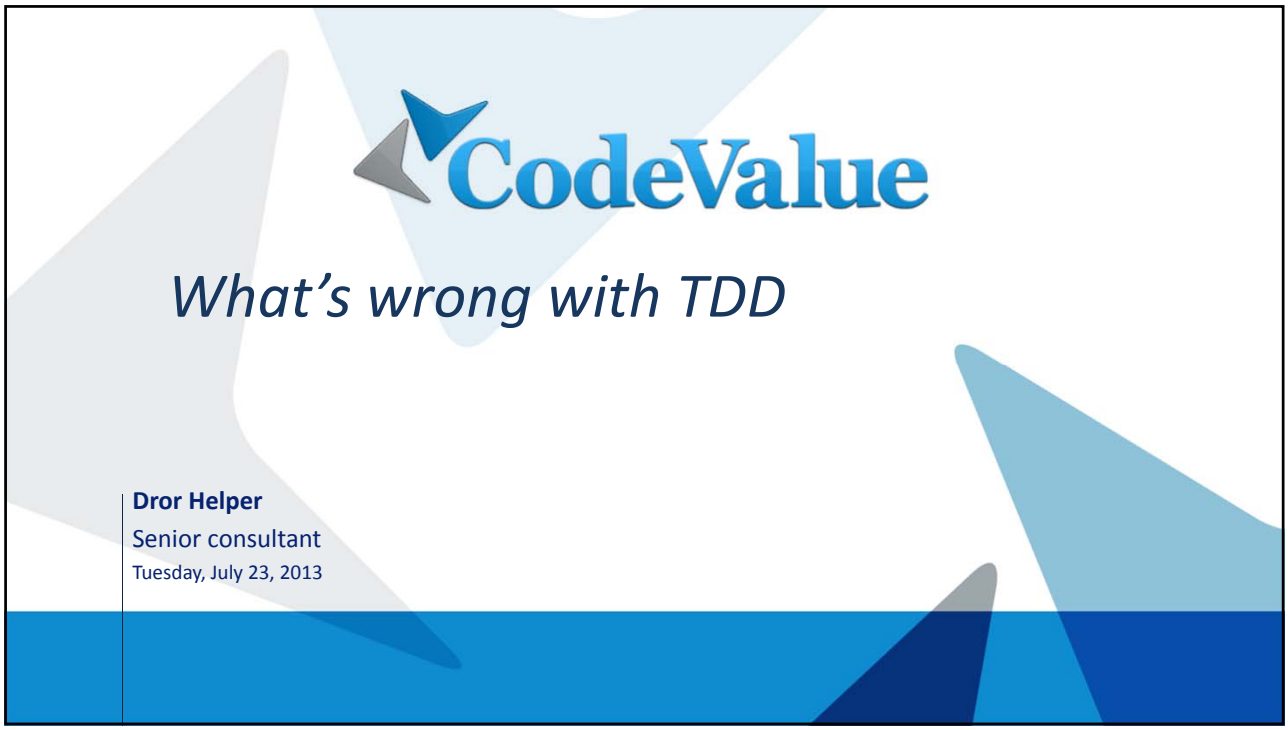


About.Me

- Software developer & Architect > 10 years
- Senior Consultant
- Agile evangelist

- Blogger - http://blog.drorhelper.com

CodeValue

What is a unit test?

A method (Code)

that

tests **specific** functionality,

Has

clear pass/fail criteria

and

runs in **isolation**

CodeValue

This is a unit test!

```
[TestMethod]
public void CheckPassword_ValidUser_ReturnTrue()
{
  bool result = CheckPassword(“user”, “pass”);

  Assert.IsTrue(result);
}
```

CodeValue

What about TDD?

Red

Write a failing test

Green

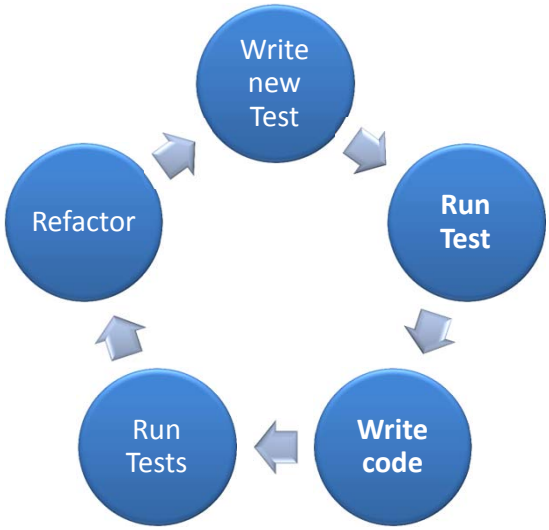
Make the test pass

Refactor

Clean duplicates and tidy up your code

CodeValue

In other words:



CodeValue

If it's so simple

WHY DOESN'T EVERYBODY DOING IT?

CodeValue

Reason #1 – it got "Tests" in it

Writing tests takes time which I don't have

or

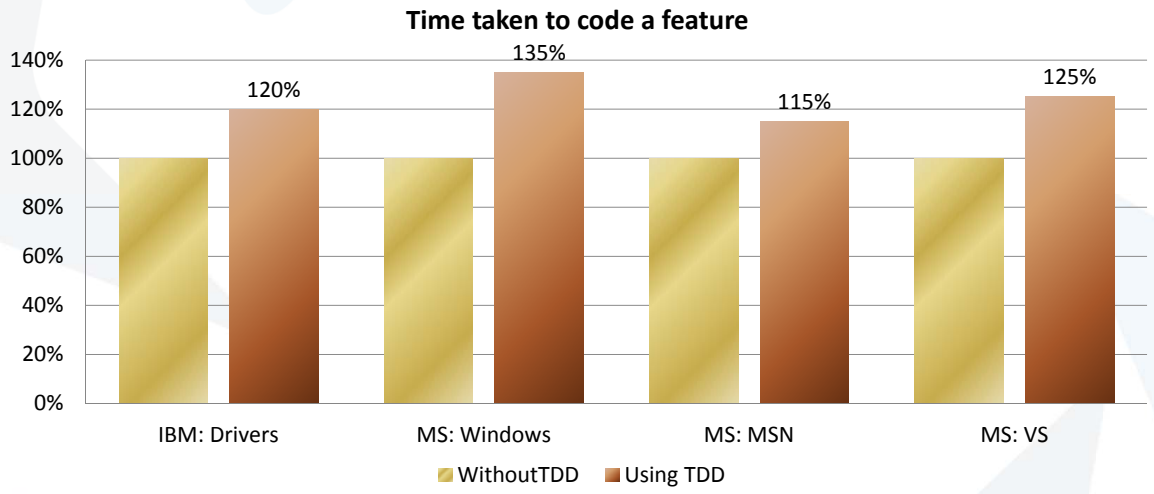
Writing tests is a waste of time

and besides

Testing is a tester's job

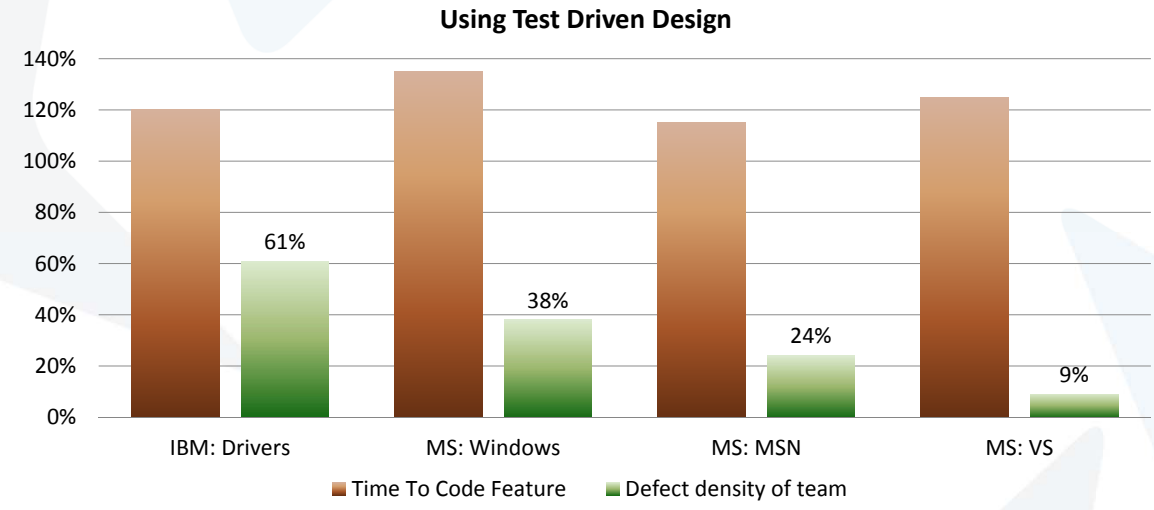
CodeValue

But it takes too much time



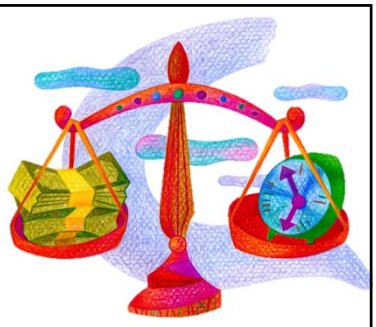
CodeValue

Is it a waste of time?



CodeValue

So do testing waste time?



`Tests costs` more time

However

`Development takes` less Time:

- Less defects
- Rarely need to invoke the debugger
- Every line has a test
- Overcome analysis-paralysis

CodeValue

Not testing is a decision

Hope Driven Development

Can you guarantee there won't be regression?

How do you know that no stupid bugs were created?

CodeValue

Reason #2 – Tests before code is wierd

Writing tests before coding is counterintuitive

How can you know what you're going to write?

Planning for failure feels wrong

CodeValue

I'll write the tests after

- Seems more logical
- Don't confuse me with new methodologies
- Just as good as tests before

And Finally - no one would ever know

It's hard to verify were written using TDD

CodeValue

TDD is not about the tests

When you write the tests after the code
You Miss the point of TDD:

Emergent Design

Only write the code you need

CodeValue

Development Driven Tests

Works – but smells funny



CodeValue

Reason # 3 – Not everything is testable

Video streaming

UI

Closed architectures

Legacy code

If I can't write tests – how can I use TDD?

CodeValue

Not everything is testable - but

You can still test business logic

Use mocking solutions

Remember - TDD is iterative

CodeValue

Reason #4 – TDD locks design

If you need to

change your code

You have to

Change your tests



CodeValue

Tests are code too

Ideally

Only requirement change (or defect)
would cause tests to break

It means

You have to know how to write robust unit tests

CodeValue

TDD is:

Requires time investment
Counter intuitive
New skill to learn
Might not work for all scenarios

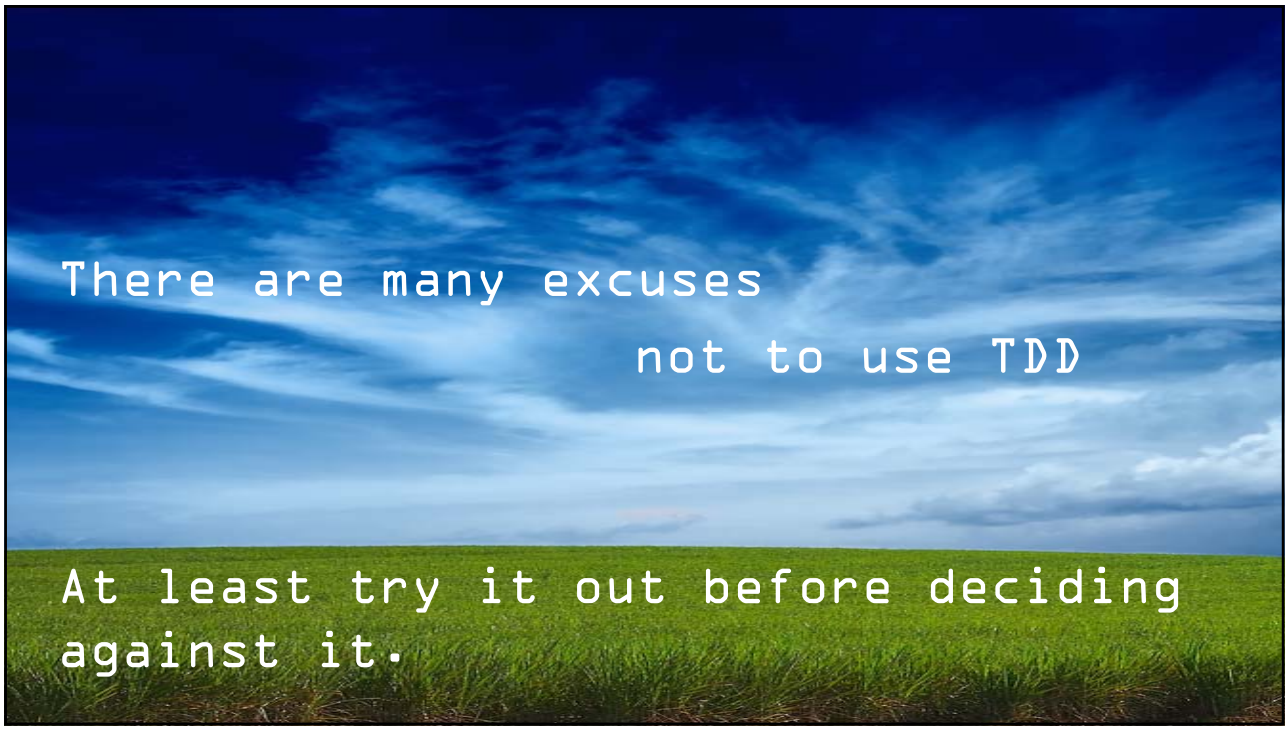
CodeValue

TDD is also:

a safety net against regression
prevents defects during development
Enabler of modular, testable code
Only write the code you need
Takes less time than DDT

If done correctly …

CodeValue



Thank You

Presenter contact details
C: 972.05.7668543
e: drorh@codevalue.net
B: blog.drorhelper.com
w: www.codevalue.net

CodeValue