

TDD

Test Driven Development

Menys temps debugant



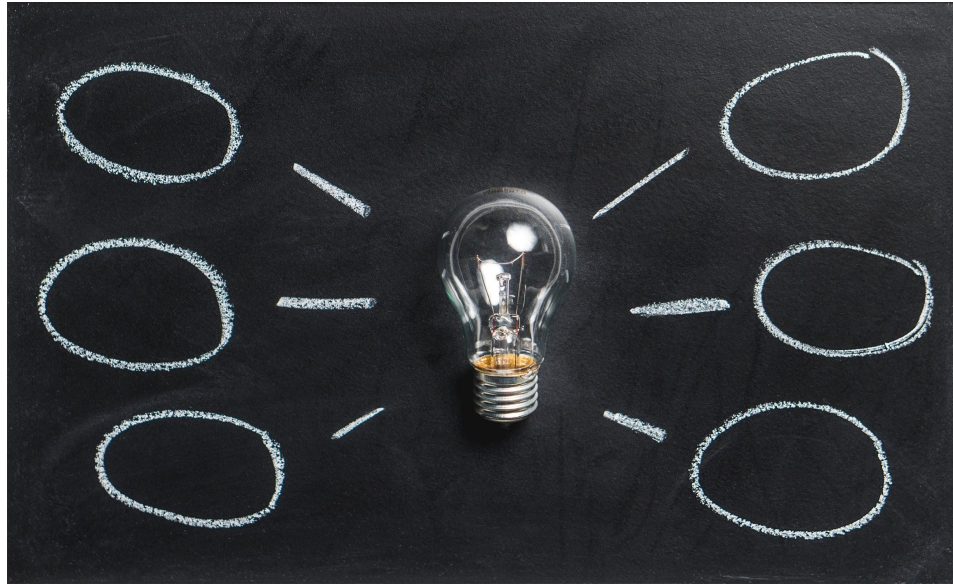
Redueix la por a fer canvis en el codi



Utilitzar els test com documentació viva



Test com feedback per prendre decisions de disseny



Les 3 regles de TDD

No està permès escriure codi que va a producció sense tenir abans una prova que falli.

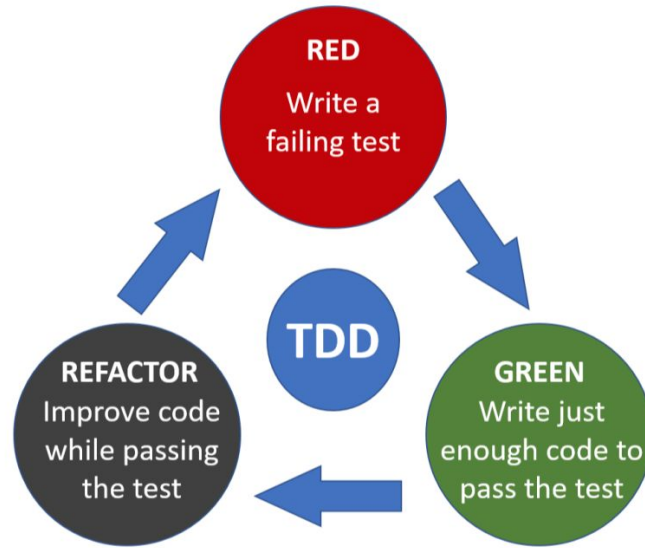
No està permès escriure més codi de prova que el necessari perquè passi la fallada.

No està permès escriure més codi que el necessari per superar la prova.

A black and white photograph of a baby crawling on a concrete curb. The baby is wearing a striped long-sleeved shirt and dark pants, and is barefoot. The baby is positioned in the center-left of the frame, leaning forward with its hands on the curb. The background is a blurred outdoor setting with a road and a grassy area. The text "BABY STEPS" is overlaid in the center of the image in a large, white, sans-serif font.

BABY STEPS

TDD FLOW



Red, Green, Refactor

Passar de vermell a verd



Fake it

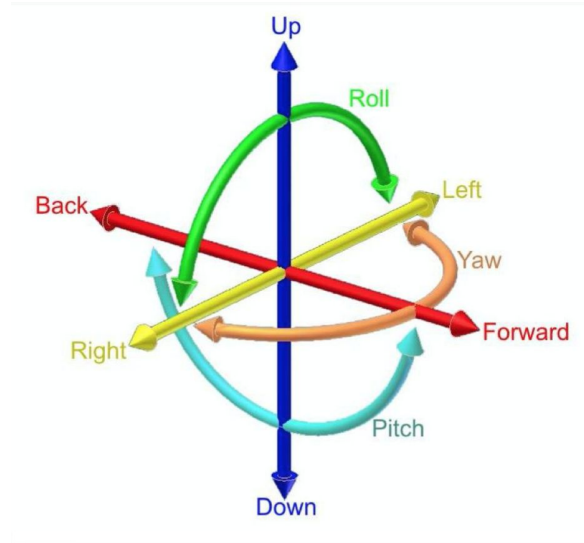
Obvious implementation (TPP)

Triangulation

Refactor



Degrees of freedom



Naming test

```
public class MyClassShould {  
    @Test  
    public void do_something() {  
    }  
}
```

Assert primero

```
public class BinaryShould {
```

```
    @Test
```

```
    public void convert_to_binary_and_merge_natural_numbers_8_and_10() {
```

```
        Binary expected = new Binary("11000100");
```

```
        Binary binaryNumber1 = new NaturalNumber(8).toBinary();
```

```
        Binary binaryNumber2 = new NaturalNumber(10).toBinary();
```

```
        Binary result = Binary.mergeInterspersed(binaryNumber1, binaryNumber2);
```

```
        Assert.assertEquals(expected, result);
```

```
    }
```

```
}
```

Given

When

Then

Arrange

Act

Assert

Kata Time



Autres concepts

Pair Programming

Mob Programming

TDD