故事是這樣的...

有一天，PM有個想法

I have a DREAM!

網站上提供
計算機的服務

功能 (Feature)

網路計算機
(Web Calculator)

匯率轉換

可以產生
大數據

一個功能，
各自表述。

1+1=2

雲端技術

要跟手機
App結合

工程計算機

使用者故事 (User Story)

As a student of primary school

In order to finish my homework

I want to do arithmetic operations

1+1=2

使用者故事
提供問題的脈絡

規格書

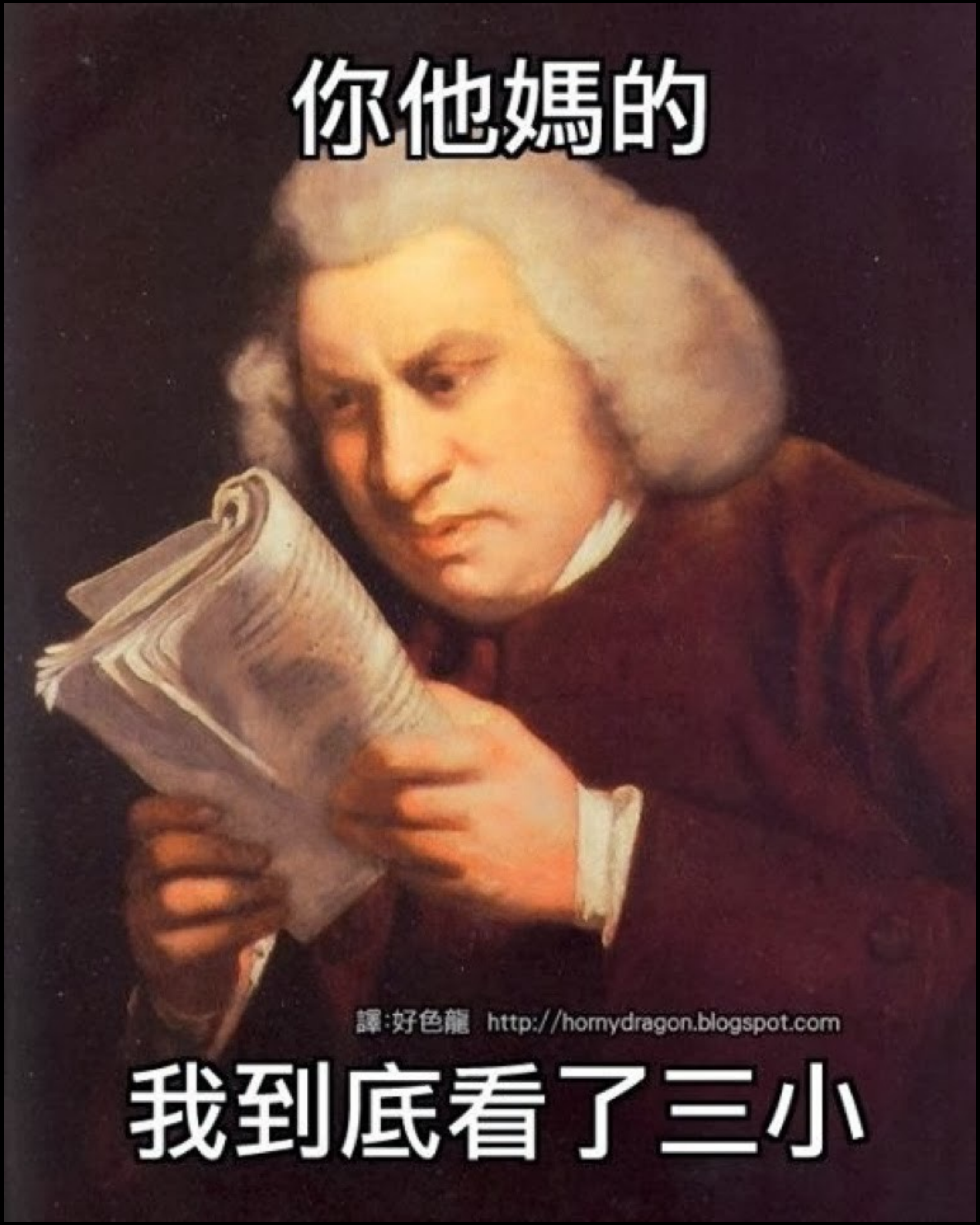
- 滿足四則運算

規格書 part2

- 滿足四則運算
  - 運算子優先順序
  - 交換律
  - 結合律
  - 分配律

規格書 part 3

- 滿足四則運算
  - 運算子優先順序: 先括號，再× ÷，後 + −
  - 交換律: $x*y = y*x \ \forall\ x,y \in S$
  - 結合律:$(x*y)*z = (x*y)*z \ \forall\ x,y,z \in S$
  - 分配律: $x*(y+z) = (x*y)+(x*z) \ \forall\ x,y,z \in S$



圖片來源 http://goo.gl/sKZQGX

能不能舉例說明？

關於行為驅動開發
(Behave Driven Development, BDD)

場景 (Scenario)

```
Scenario Outline: do simple operations
    Given I enter <expression>
     When I press "=" button
     Then I get the answer <answer>

    Examples:
        | expression      | answer        |
        | 3 + 2           | 5             |
        | 3 - 2           | 1             |
        | 3 * 2           | 6             |
        | 3 / 2           | 1.5           |
        | 3 +-*/ 2        | Invalid Input |
        | hello world     | Invalid Input |
```

加法/乘法交換律

```
Scenario Outline: satisfy commutative property
     When I enter <expression1> first
      And I enter <expression2> again
     Then I get the same answer

    Examples:
        | expression1    | expression2    |
        | 3 + 4          | 4 + 3          |
        | 2 * 5          | 5 * 2          |
```

加法/乘法結合律

```
Scenario Outline: satisfy associative property
     When I enter <expression1> first
      And I enter <expression2> again
     Then I get the same answer

    Examples:
        | expression1   | expression2   |
        | (2 + 3) + 4   | 2 + (3 + 4)   |
        | 2 * (3 * 4)   | (2 * 3) * 4   |
```

乘法左/右分配律

```
Scenario Outline: satisfy distributive property
     When I enter <expression1> first
      And I enter <expression2> again
     Then I get the same answer

    Examples:
        | expression1    | expression2    |
        | 2 * (1 + 3)    | (2*1) + (2*3) |
        | (1 + 3) * 2    | (1*2) + (3*2) |
```

RD: 為什麼不測試這個?

```
Scenario Outline: parse an expression
    Given I enter <expression>
     When I press "=" button
     Then I get an <array>

    Examples:
        | expression     | array           |
        | 1+2            | ['1','+','2']   |
        | 1*2            | ['1','*','2']   |
```

關於測試

我說的其實是......

“Because designing the technical solution is not the purpose of the specification, you should focus only on writing scenarios that relate to the business rules.”

- Executable Specification with Scrum

QA:
驗收測試，讓專業的來

執行測試

```
$ python manage.py behave --dry-run
…(略)

You can implement step definitions for undefined steps with these
snippets:

@given(u'I enter "3+2"')
def step_impl(context):
    raise NotImplementedError(u'STEP: Given I enter "3+2"')

@when(u'I press "=" button')
def step_impl(context):
    raise NotImplementedError(u'STEP: When I press "=" button')

@then(u'I get the answer "5"')
def step_impl(context):
    raise NotImplementedError(u'STEP: Then I get the answer "5"')
```

重構步驟

複製貼上，修修改改

```
@given(u'I enter {expr}')
def step_impl(context, expr):
    raise NotImplementedError(u'STEP: Given I enter {expr}')

@when(u'I press "=" button')
def step_impl(context):
    raise NotImplementedError(u'STEP: When I press "="
button')

@then(u'I get the answer {answer}')
def step_impl(context, answer):
    raise NotImplementedError(u'STEP: Then I get the answer
{answer}')
```

…(略)

執行測試

```
$ python manage.py behave

Creating test database for alias 'default'...
Feature: Web calculator # features/calc.feature:3
  As a student
  In order to finish my homework
  I want to do arithmatical operations
  Scenario Outline: do simple operations -- @1.1
    Given I enter 3 + 2
      Traceback (most recent call last):
        ...(略)
    NotImplementedError: STEP: Given I enter {expr}

    When I press "=" button
    Then I get the answer 5
```

溫馨提示：還沒拿掉 NotImplementError

重構步驟

```
from calc.calculator import Calculator

@given(u'I enter {expr}')
def step_impl(context, expr):
    context.expr = expr

@when(u'I press "=" button')
def step_impl(context):
    calc = Calculator()
    context.answer = calc.evalString(context.expr)

@then(u'I get the answer {answer}')
def step_impl(context, answer):
    assert context.answer == answer
```

假裝類別存在

假裝有個方法

執行測試

```
$ python manage.py behave

Creating test database for alias 'default'...
Exception ImportError: No module named 'calc.calculator';
'calc' is not a package
Traceback (most recent call last):
  ...(略)
  File "/home/vagrant/myWorkspace/demo/features/steps/
calc.py", line 1, in <module>
    from calc.calculator import Calculator
ImportError: No module named 'calc.calculator'; 'calc' is not
a package
```

溫馨提示：還沒實作 calc.calculator

拉出介面

```
#file: calc/calculator.py

class Calculator:

    def evalString(self, string):
        return 0
```

回傳的預設值

執行測試

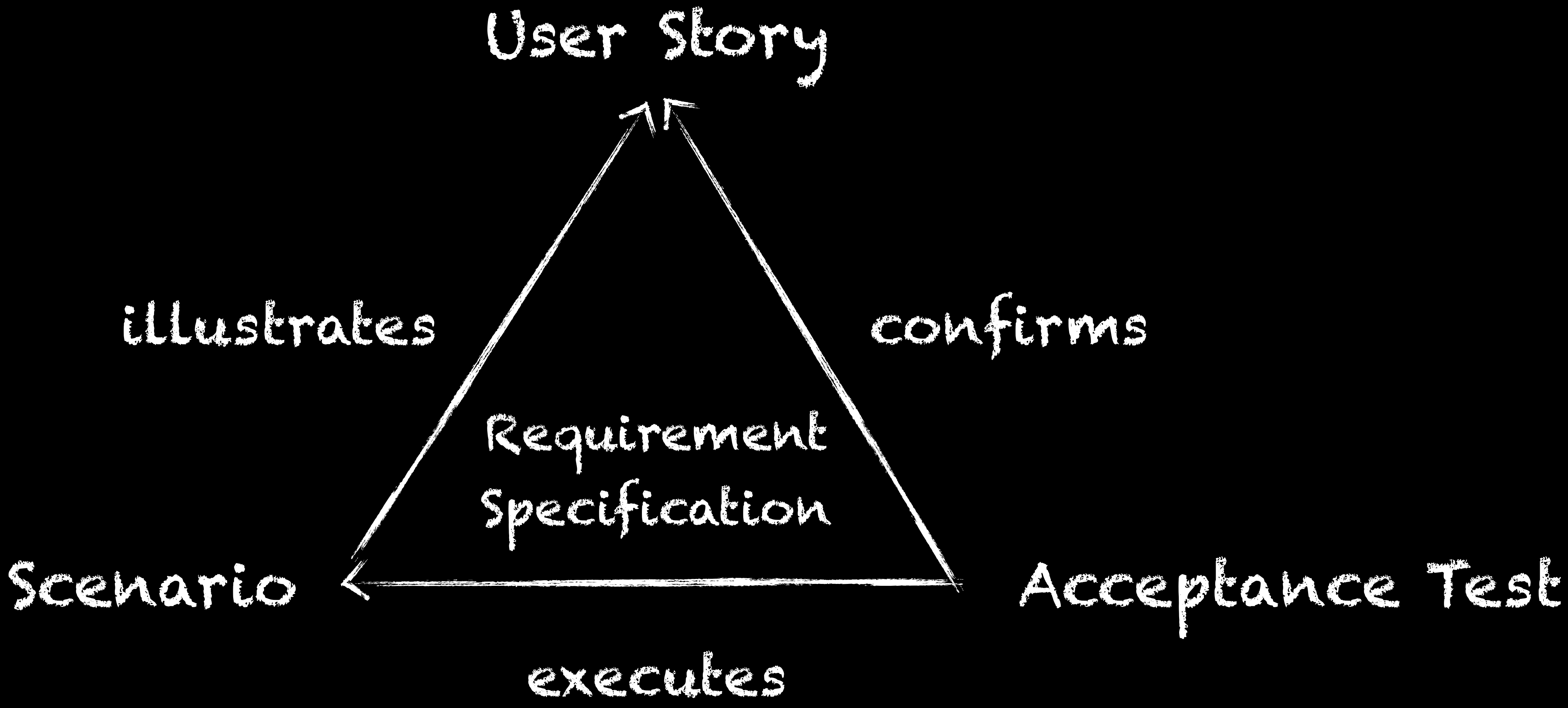
```
$ python manage.py behave

Creating test database for alias 'default'...
Feature: Web calculator # features/calc.feature:3
  As a student
  In order to finish my homework
  I want to do arithmatical operations
  Scenario Outline: do simple operations -- @1.1
    Given I enter 3 + 2
    When I press "=" button
    Then I get the answer 5
      Traceback (most recent call last):
        ...(略)
        File "features/steps/calc.py", line 19, in step_impl
          assert context.answer == answer
      AssertionError
```

QA: 接下來就是 RD 的事了

關於測試

我說的其實是……



通過場景轉換成驗收測試，
規格變成一份可執行的活文件。

有些人熱愛中華文化

圖片來源 http://goo.gl/BQkP82

說中文也行

場景大綱：　做簡單的運算
　　假設<　我輸入<expression>
　　　當<　我按下等號按鈕
　　那麼<　我得到的答案是<answer>

例子：

expression	answer
3 + 2	5
3 - 2	1
3 * 2	6
3 / 2	1.5
3 +-*/ 2	Invalid Input
hello world	Invalid Input

執行測試

```
$ python manage.py behave --dry-run --include
zh_calc

功能: 網頁計算機 # features/zh_calc.feature:2
  身為一個學生
  為了完成家庭作業
  我想要做算術運算
  場景大綱: 做簡單的運算 -- @1.1
    假設 < 我輸入3 + 2
    當 < 我按下等號按鈕
    那麼 < 我得到的答案是5

  …(略)
```

關於測試

我說的其實是......

BDD 不只是測試框架，

更是**溝通需求**的哲學。