

# 寫完實例化需求，然後呢？

**Jamis Liao**

(<https://dotblogs.com.tw/jamis/1>)



# ME

Jamis

2016微軟最有價值專家

## Tag

Asp.NetMVC' C#' DevOps' 軟體開發流程' 系統架構'

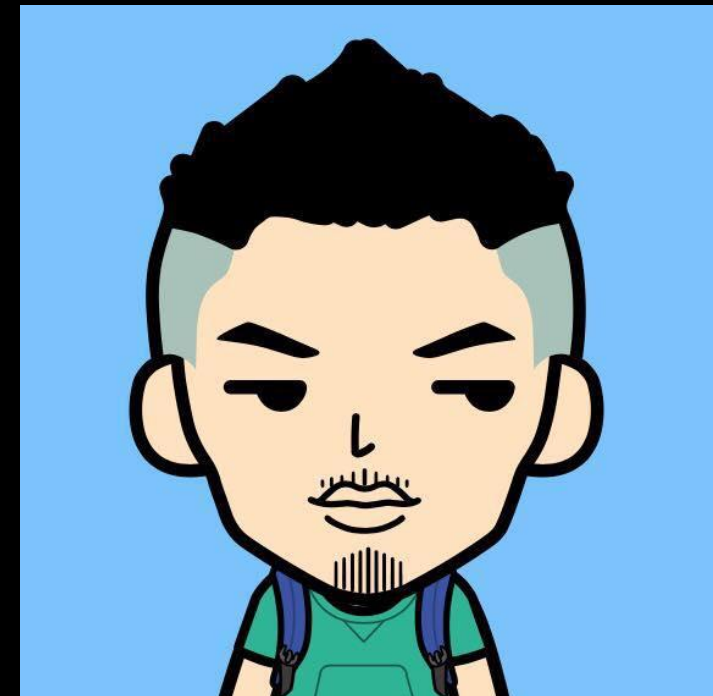
Cloud Architecture

## Blog

天橋下的說書人

## Facebook

Jamis Liao



**Microsoft®**  
Most Valuable  
Professional

實例化需求

不管是

PM、PO、Scrum Master、工程師

應該都會遇過以下狀況

需求文件不知道如何寫工程師才看得明瞭  
需求文件看完但完全不知道要做什麼

功能不知道是否有作完整

功能不知道該如何測試



又或者是有這樣的症狀

好不容易約個時間來討論需求文件，但是……………



你這是專程來吵架的？  
Do you want to quarrel with me?

好不容易坐下來討論需求文件了



才討論到一半就變成

乐视

再说一句

最後就.....





不錯！少林寺十八銅人！  
Right! Eighteen Brassmen of Shaolin Monastery!

希望透過改變撰寫的方式  
來改善

使用自然語言

透過團隊的協同合作

具體的案例  
使用的數據  
發生的時機  
實際的結果  
驗證的項目

改善

抽象的需求  
不確定的範圍  
未知的測試項目

Cucumber



What  
is  
Cucumber

Cucumber是一款頗受贊譽的針對**行為驅動開發 (BDD)**而設計的**測試工具**。讓人們用**最少的技術術語來描述系統需求**。通過純文本來描述他們需要系統如何執行的具體實例。用簡潔的描述，比如：“在Bob的儲蓄帳戶上應該有5000美元存款”。用純文本作為編好的代碼需求說明。



Aslak Hellesøy(2009年11月23日 InfoQ)

Cucumber首先是一種**協作工具**，它旨在讓團隊中的所有成員之間達成共識。Cucumber的特性編寫應當早于具體用代碼實現這些特性。當你使用BDD方式編寫實例的時候，回歸測試會自然地成為一種副帶的結果，但測試本身並不是這個活動的目的。



Aslak Hellesøy( 2015年4月2日 InfoQ)

簡單來說

Cucumber是個協作工具

基於Behavior-Driven Development

搭配套件可以將需求描述  
轉成測試程式

**Scenario:** 使用VIP帳號購買五本書籍，不需運費

**Given** 購物車中有五本書

**When** 結帳計算運費時

**Then** 運費應該為 0



# Gherkin

- Gherkin is the language that Cucumber understands.
  - Business Readable,
  - Domain Specific Language
  - Describe software's behaviour without detailing how that behaviour is implemented.

# Gherkin關鍵字

- Feature
- Background
- Scenario
- Step
  - Given
  - When
  - Then
  - And
  - But

- 功能
- 背景
- 場景
- 步驟
  - 架設
  - 當
  - 那麼
  - 並且
  - 但是

**Scenario:** 使用VIP帳號購買五本書籍，不需運費

**Given** 購物車中有五本書

**When** 結帳計算運費時

**Then** 運費應該為 0

# Cucumber通用於單元測 試與整合測試

# 單元測試

**Scenario:** 手續費於交易中計算扣款，日結算時計算手續費

**Given** 交易資料

<i>ReceiverId</i>	<i>TransAmount</i>	<i>Fee</i>	<i>FinalAmount</i>	<i>TransDate</i>
123	100	2	98	2016-11-07
123	123	2	121	2016-11-07
123	111	2	109	2016-11-07
123	224	3	221	2016-11-07

**And** 店家手續費設定資料

<i>Fee</i>	<i>FeeCalculateCycle</i>
0.015	1

**When** 計算手續費時

**Then** 手續費金額應該為

<i>Fee</i>	<i>TransAmount</i>
9	558

# 整合測試

```
#language: zh-TW
```

## @圖片服務整合測試

## 功能： 圖片服務整合測試

提供使用者儲存或取得圖片

## 透過圖片服務同時存取多個容器檔案

## 背景：

假設 `AwsAccessKey` 為XXXXXXXXXXXXXXXX

[illegible]

並且 BucketName為AgileTourTaipei.test

並且 清空Bucket

**並且** 無群組機制圖片儲存根目錄為C:\Temp\ImageTest

**並且 清除無群組機制圖片根目錄內容**

假設 S3ImageService的模組註冊順序為1

假設 FileImageService的模組註冊順序為2

並且 建立服務

# 整合測試

場景：儲存檔案，檔案儲存後將回傳檔案路徑 - FilePath (來源路徑, 目標路徑)

假設 本地端產生一個檔案，檔案名稱為test.jpg

當 儲存圖片，檔案名稱為test.jpg

那麼 檔案應存在S3的位置test.jpg

那麼 本地目錄存在檔案test.jpg

那麼 各儲存裝置的檔案路徑

ImageProvider	ImagePath	
FileImage	C:\Temp\ImageTest\test.jpg	
S3Image	test.jpg	

# Cucumber implementations



Ruby/JRuby



JRuby (using Cucumber-JVM)



Java



Groovy



JavaScript



JavaScript (using Cucumber-JVM and Rhino)



Clojure



Gosu



Lua



.NET (using SpecFlow)



PHP (using Behat)



Jython



C++



Tcl



持續整合

如果是人工整合

Pull Code

Build Code

Testing

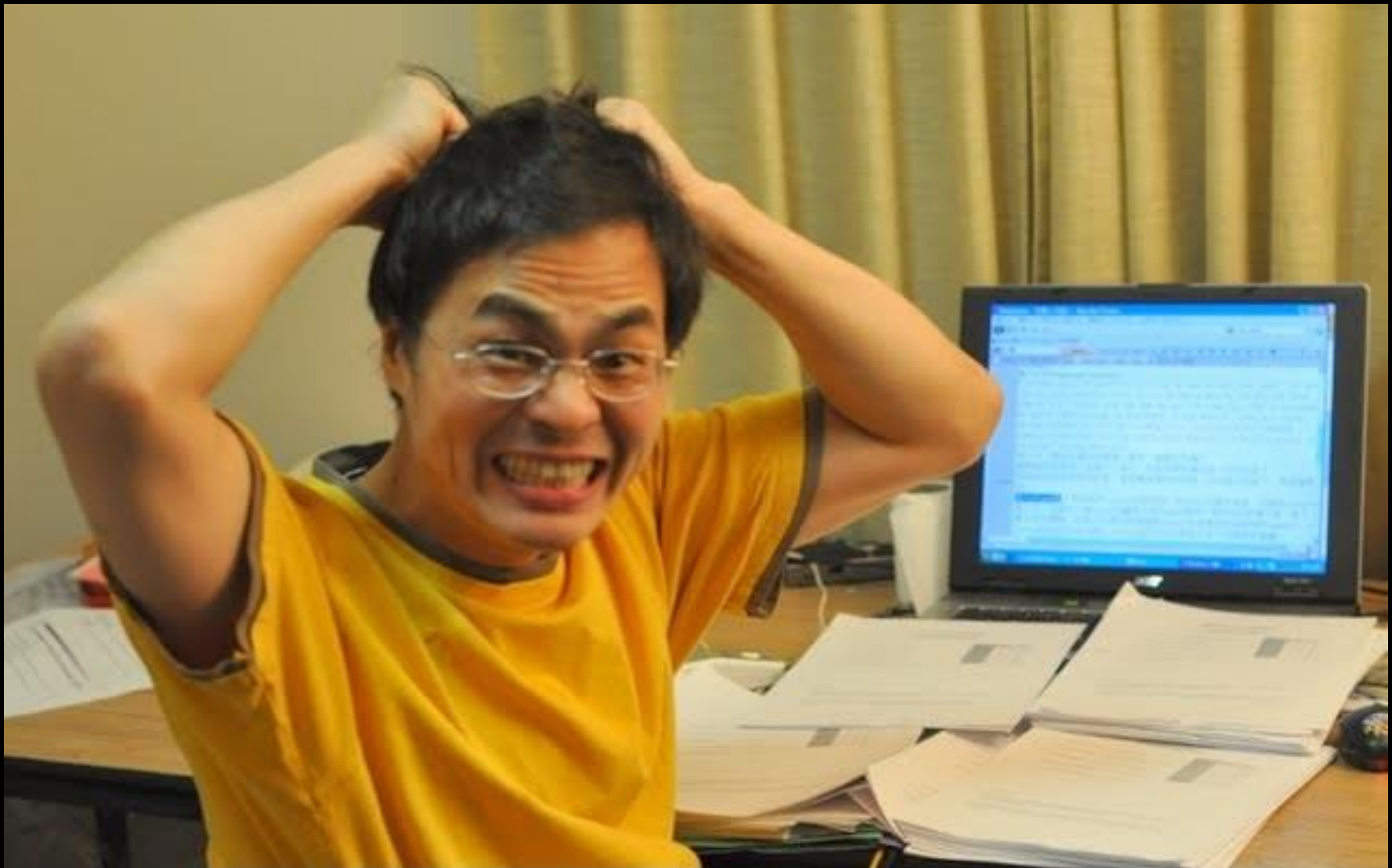
Pull Code

Build Code

Testing



日復一日



<http://tienlan.pixnet.net/blog/post/33427954-%E2%98%85%E2%98%85%E8%80%83%E5%8D%B7%E6%94%B9%E5%88%B0%E5%BF%AB%E7%99%BC%E7%98%8B%E2%98%85%E2%98%85>

所以你需要一位大叔

(這絕對不是業配文.....)



幫你做

# CI Server

- CI Server的精神就是，反覆' 重複' 不斷的持續進行
- 融入了我們的日常開發之中隨時存在
- 我們不需要花太多的時間關注他，但是他隨時存在著

# Jenkins

- 易於安裝
- 設定簡單
- 整合RSS/Email通知
- 支援分散式架構
- 大量的套件

該如何開始與設定？

安裝擴充套件

人工如何做，就如何  
設定Jenkins



只是讓Jenkins執行原本人  
工執行的指令

# Command類型

- 還原專案套件
- 判斷資料夾是否存在
- 產生資料夾
- 判斷檔案是否存在
- 刪除檔案
- 執行 .exe

.exe位置 指令(選) -參數名稱(選) -參數(選)  
-參數名稱(選) -參數(選)

.\specflow.exe" generateAll "...\\AgileTourTaipei2016.csproj  
/force /verbose

Pickles

可以將測試案例轉換成文件

還可以順便結合測試結果

# Test Result Format

- Nunit
- Nunit3
- Xunit
- Xunit2
- Mstest
- Cucumberjson
- Specrun
- stest



# Input

- Feature

## Example Output

- HTML site with search capabilities
- Static HTML site
- JSON raw datas
- Word (.docx)
- Excel

每次當Jenkins Build專案時，  
自動Re Build文件

文件永遠是最新的狀態

# 參考資料

- Sample Github
  - <https://github.com/jamisliao/AgileTourTaipei2016>
- Specflow
  - <https://github.com/techtalk/SpecFlow>
- Pickles
  - <http://www.picklesdoc.com/>
  - <https://github.com/picklesdoc/pickles>
- Jenkins
  - <http://ithelp.ithome.com.tw/articles/10109773>
- InfoQ
  - <https://goo.gl/GcM7gf> (簡中)
  - <https://goo.gl/JsUXkR> (原文)
  - <https://goo.gl/fMqr8A> (簡中)
  - <https://goo.gl/vp31VG> (原文)