

# CH4.測試策略的實作

曾義格、江少傑

# Outline

- 引言
- 測試的分類
- 考量

# 引言

- 沒有測試過的東西，就先假設它是壞掉的
- 從流程的一開始就進行測試（縮短回饋時間）
- 需求的definition of done

# 測試的分類

業務導向 vs 技術導向

支援日常開發活動 vs 評估性質

# 驗收測試

- 在User story中定義驗收條件（接近自然語言）
- 需要開發人員再撰寫測試程式（程式語言）
- 透過工具將驗收條件對應到測試程式
  - Cucumber
  - Spock
  - Jbehave
- 可以透過工具執行
  - UI測試？
  - 只測api？

# 單元測試

- 完全由開發人員撰寫
- 獨立於外部相依
- 可以透過工具自動執行

# 整合測試

- 驗證系統之間的互動
- 需要模擬外部系統
  - Mock server

# 其他

- 探索性測試
  - 驗證 vs 測試
  - 運用domain knowhow，在既定的驗證路徑以外，找出系統可能出現問題的地方，去測試是否真的有問題
  - 找出更有效率的驗證方式
- A/B測試

```
given:  
  int a=1  
  int b=2  
when:  
  result=a+b  
then:  
  assert result==3
```



考量

# 理想與現實

- 系統的可測試性
- 資源
  - 撰寫
  - 執行
  - 維護
- 測試程式的涵蓋率、逼真度、可維護性
- 管理缺陷清單
- 自動化的代價

# 進行中的專案

- 盡快開始寫測試
- 至少涵蓋Happy path
- 自動化執行

# 既有系統

- 針對被發現的缺陷寫測試
- 針對新需求寫測試
- 提高可測試性
- Stop digging!

# 流程

- 測試團隊依據user story撰寫測試腳本
- 在開發初期讓測試團隊參與，直接跟客戶討論測試腳本

# 管理缺陷清單

- 可視化
- 與待開發需求一起排定優先順序

感恩