

# Angular - Test

by Leo

# 大綱

- 為什麼要寫測試
- 目的
- 種類
- 心法
- 覆蓋率
- TDD vs ATDD

# 為什麼要寫測試

- 程式發生錯誤，怎麼模擬出當時狀況及Debug?
- 改了這個function，會不會導致別的程式掛掉?
- 到底測過了哪些功能，有沒有漏掉的還沒測?

# 目的

- 能快速提供反饋
- 提昇程式碼品質
- 節省測試與除錯時間
- 整體專案時程縮短

# 種類

- 單元測試
- 整合測試
- 驗收測試

# 單元測試

- 什麼是單元測試？
  - 用來模擬外部如何使用測試目標物件，驗證其**行為**是否符合預期
  - 從程式最小的功能開始
  - 單元指的是一個類別或一個模組

# 整合測試

- 什麼是整合測試？
  - 整合測試是將兩個以上的類別做整合，並且測試它們之間的運作**關係**是不是正確的

# 驗收測試

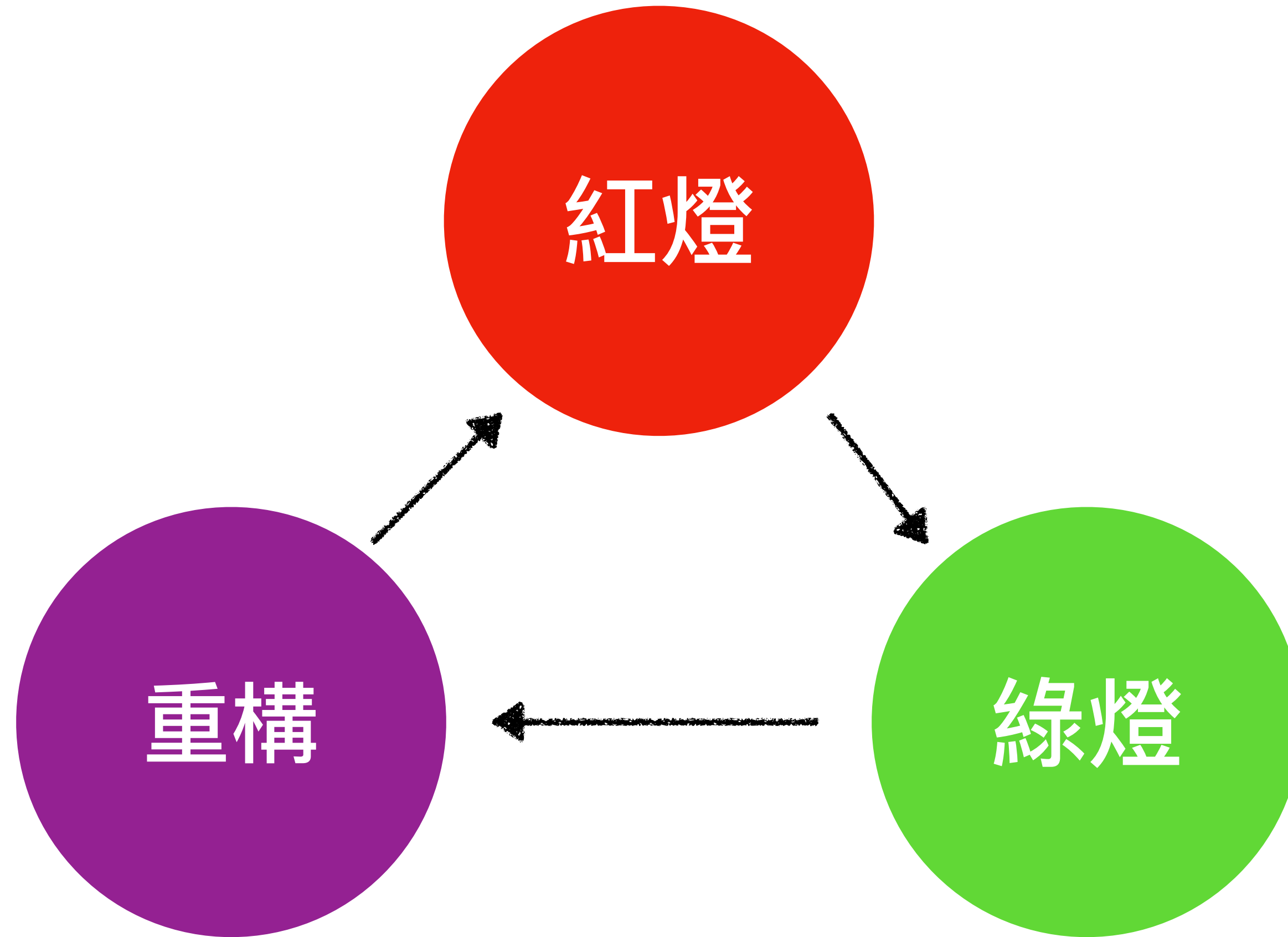
- 什麼是驗收測試？
  - 系統行為與功能面的規範
  - 用來說明某一個user story
  - 從**使用者的角度來檢視**（是否符合使用者的期望）



# 心法

- 在編寫某個功能的代碼之前先編寫測試代碼，然後只編寫使測試通過的功能代碼
- 所有的實現都是測試「逼」出來的，所有的實現代碼都是為了讓測試通過而編寫的
- 先以調用方的角度來調用這塊代碼，並且從調用方的角度說出所期望的結果。
- 編寫測試時就僅僅關注測試，不想去如何實現

# 心法



# 心法

- 絕不跳過重構
- 儘快變綠
- 出錯後放慢腳步

# 覆蓋率

- 測試程式碼的涵蓋範圍
- 請把覆蓋率的數字當作一個健康指標，用來檢查：
  - 重要的Scenario有沒有涵蓋多一點情境？
  - 現在有哪些程式碼沒有被測過？
  - 發生測試失敗時，失敗的原因點是否有被測試案例涵蓋到？

# TDD vs ATDD

- TDD：測試驅動開發 (Test-Driven Development)
- ATDD：驗收測試驅動開發 (Acceptance Test Driven Development)

# TDD vs ATDD

- 軟體開發過程中最常見的問題：
  - 用戶想要的功能沒有開發
  - 開發的功能並非用戶想要
  - 用戶和開發人員所說語言不同

# TDD vs ATDD

TDD

ATDD



單元測試

整合測試

驗收測試



# 參考資料與延伸閱讀

- 凌群電子報
- 菜鳥階段-測試很重要
- 各種測試介紹
- 淺談TDD、BDD與ATDD軟體開發
- 測試驅動開發——我們的不僅僅是「質量」
- [如何提升系統品質-Day24]測試 - Code Coverage



# 參考資料與延伸閱讀

- ATDD 測試觀念及手法
- Jasmine & Protractor 簡介
- 如何寫 Angular 的測試之學習心得