測試驅動開發
Test-Driven Development

準備開發環境

```
# sudo apt update
# sudo apt install -y python3 python3-pip vim git
# pip3 install -y pytest pytest-cov
```

測試大亂鬥

測試知多少？

黑箱測試

可用性測試

探索性測試

回歸測試

安全性測試

效能測試

驗收測試

灰箱測試

壓力測試

系統測試

負載測試

整合測試

功能測試

白箱測試

單元測試

邊界測試

測試方法

- 黑箱測試
  - 可用性測試
  - 探索性測試
  - 回歸測試
  - 安全性測試
  - 效能測試
  - 驗收測試
- 灰箱測試
  - 壓力測試
  - 系統測試
  - 負載測試
  - 整合測試
  - 功能測試
- 白箱測試
  - 單元測試
  - 邊界測試

測試目的

黑箱測試　可用性測試　探索性測試　回歸測試

安全性測試　效能測試　驗收測試

灰箱測試　壓力測試　系統測試

負載測試　整合測試　功能測試

白箱測試　單元測試　邊界測試

非功能測試

黑箱測試

可用性測試

探索性測試

回歸測試

安全性測試

效能測試

驗收測試

灰箱測試

壓力測試

系統測試

負載測試

整合測試

功能測試

白箱測試

單元測試

邊界測試

測試進程(功能相關)

黑箱測試

可用性測試

探索性測試

回歸測試

安全性測試

效能測試

驗收測試

灰箱測試

壓力測試

系統測試

負載測試

整合測試

功能測試

白箱測試

單元測試

邊界測試

測試策略

黑箱測試

可用性測試

探索性測試

回歸測試

安全性測試

效能測試

驗收測試

灰箱測試

壓力測試

系統測試

負載測試

整合測試

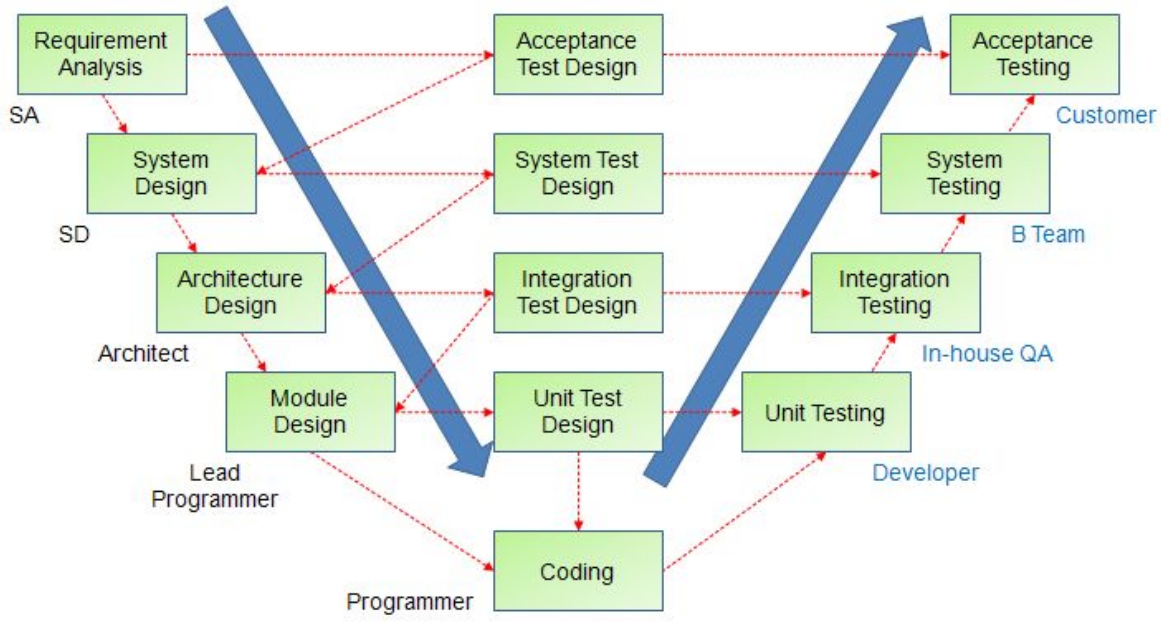
功能測試

白箱測試

單元測試

邊界測試

V Model


Requirement Analysis
SA
System Design
SD
Architecture Design
Architect
Module Design
Lead Programmer
Coding
Programmer
Acceptance Test Design
System Test Design
Integration Test Design
Unit Test Design
Unit Testing
Developer
Integration Testing
In-house QA
System Testing
B Team
Acceptance Testing
Customer


軟體品質是測出來的！

FIRST 原則

- ✓ Fast：快速。
- ✓ Independent：獨立。
- ✓ Repeatable：可重複。
- ✓ Self-Validating：可反應驗證結果。
- ✓ Timely：及時。

FIRST 原則

- ✓ Fast：快速。
- ✓ Independent：獨立。
  - ✓ 外部相依性必須為零，舉凡檔案、網路、其他類別都不能影響測試的結果。
  - ✓ 測試案例(TestCase) 之間的相依性也應該要為零。
- ✓ Repeatable：可重複。
- ✓ Self-Validating：可反應驗證結果。
- ✓ Timely：及時。

FIRST 原則

- ✓ Fast：快速。
- ✓ Independent：獨立。
- ✓ **Repeatable：可重複。**
  - ✓ **無論何時、何地，測試的結果不能改變。**
- ✓ Self-Validating：可反應驗證結果。
- ✓ Timely：及時。

FIRST 原則

- ✓ Fast：快速。
- ✓ Independent：獨立。
- ✓ Repeatable：可重複。
- ✓ Self-Validating：可反應驗證結果。
  - ✓ 每一個測試案例都真能驗證某一件事。
  - ✓ 單元測試不論成功或失敗，都應該要從測試的 reporting 直接瞭解其意義或失敗原因。
  - ✓ 過程不應該需要手動操作，才能判斷結果是否正確。
- ✓ Timely：及時。

FIRST 原則

- ✓ Fast：快速。
- ✓ Independent：獨立。
- ✓ Repeatable：可重複。
- ✓ Self-Validating：可反應驗證結果。
- ✓ Timely：及時。
  - ✓ 單元測試和程式碼應同時存在。
  - ✓ 不要相信測試程式碼有可能之後會補齊，那是不可能發生的。

3A 原則

為了增加測試程式碼的可讀性

1. Arrange (初始、期望結果)
   - ✓ 初始化目標物件和相依物件
   - ✓ 設定物件之間的互動方式
   - ✓ 指定功能執行後的預期結果
2. Act (實際呼叫)
   - ✓ 執行待測試的功能。
3. Assert (驗證)
   - ✓ 驗證結果是否符合預期。

 # Giving

 # When

 # Then

覆蓋率

函式(function)覆蓋率
指令(statement)覆蓋率

判斷(decision)覆蓋率
條件(condition)覆蓋率
條件/判斷覆蓋率

修改條件判斷覆蓋率

A等級軟體

線性代碼序列和跳轉(JCSAJ)覆蓋率

先求有

再求好

TDD

- ✓ TDD 是一種開發方法，不是測試
  - ✓ 將規格以可執行的「測試案例」來表達
  - ✓ 測試先行

- ✓ TDD 3階段
  1. Red
  2. Green
  3. Refactor

- ✓ 單元測試 / 自動化測試

TDD 的迷思

- ✓ 測試案例是測試人員在寫的
- ✓ 測試案例寫愈多愈好
- ✓ TDD不需要軟體設計
- ✓ TDD / Agile 可以加快開發速度
- ✓ 程式不會寫，學TDD就對了

測試是開發中神聖不可分割的一環

讓你的程式碼具備應變的能力

TDD不會提升程式能力
(請右轉 OOAD / Design Pattern)

TDD 能夠...

- ✓ 讓程式碼符合需求
- ✓ 避免過度設計
- ✓ 改善 API 的設計與可用性
- ✓ 避免邏輯錯誤
- ✓ 案例即文件
- ✓ 確保問題已修正
- ✓ 不怕改壞程式碼
- ✓ 提升程式品質