



## Part VI 軟體測試

李信杰 副教授

國立成功大學資訊工程學系



# Bug-free software?

If you build it, they will come



Yeah, I'm just  
writing the code now.





# 軟體測試重要性

## □台灣彩卷系統當機

- 2007年，台灣彩券頭彩獎金累積超過十億，造成民眾湧入彩券行搶購彩券。因彩券發售系統未經適當的壓力測試(Stress testing)，以至於無法負荷大量的購票。

## □法國雅利安五號銷毀

- 1996年，法國雅利安五號(Ariane 5)運載火箭才升空不到一分鐘，就自動銷毀，損失總金額估計高達數十億元，且雅利安計畫因此停滯數個月。調查委員會說明失敗的過程為：「在主引擎點火後37秒，導向與高度的資訊完全喪失」；而造成此失敗的原因是：「內部參考系統軟體的規格與設計錯誤」。某一段程式碼要將64位元的數字裝填到16位元的空間，使得運算過程產生溢位現象(Overflow)。



# 軟體測試重要性

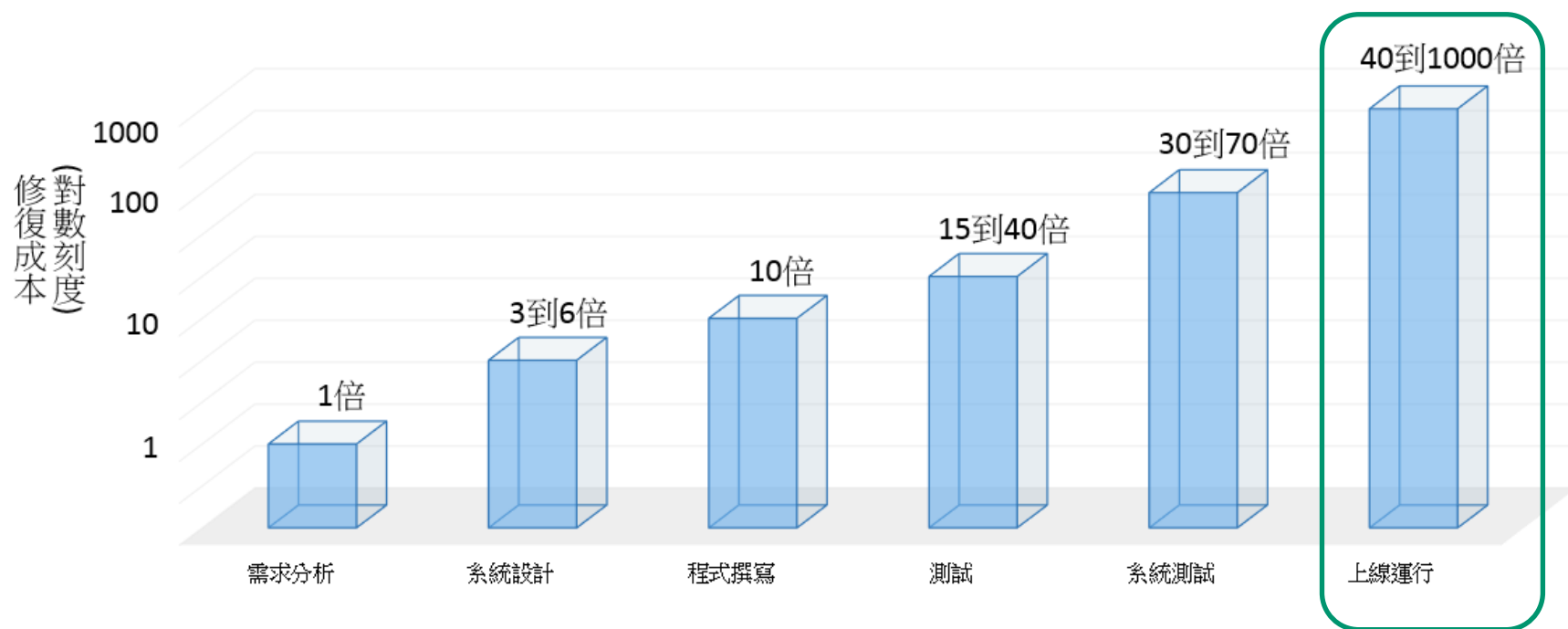
□丹麥107歲人瑞收到小學入學通知：

- 丹麥曾有一位107歲人瑞收到一封電腦統一印製的信件，內容是小學入學通知書。經追查後發現，當初設計這套程式的工程師沒考慮到百歲人瑞，僅以兩位數儲存年齡，因此107歲在系統中被視為7歲，所以電腦自動印製這位人瑞的入學通知書。



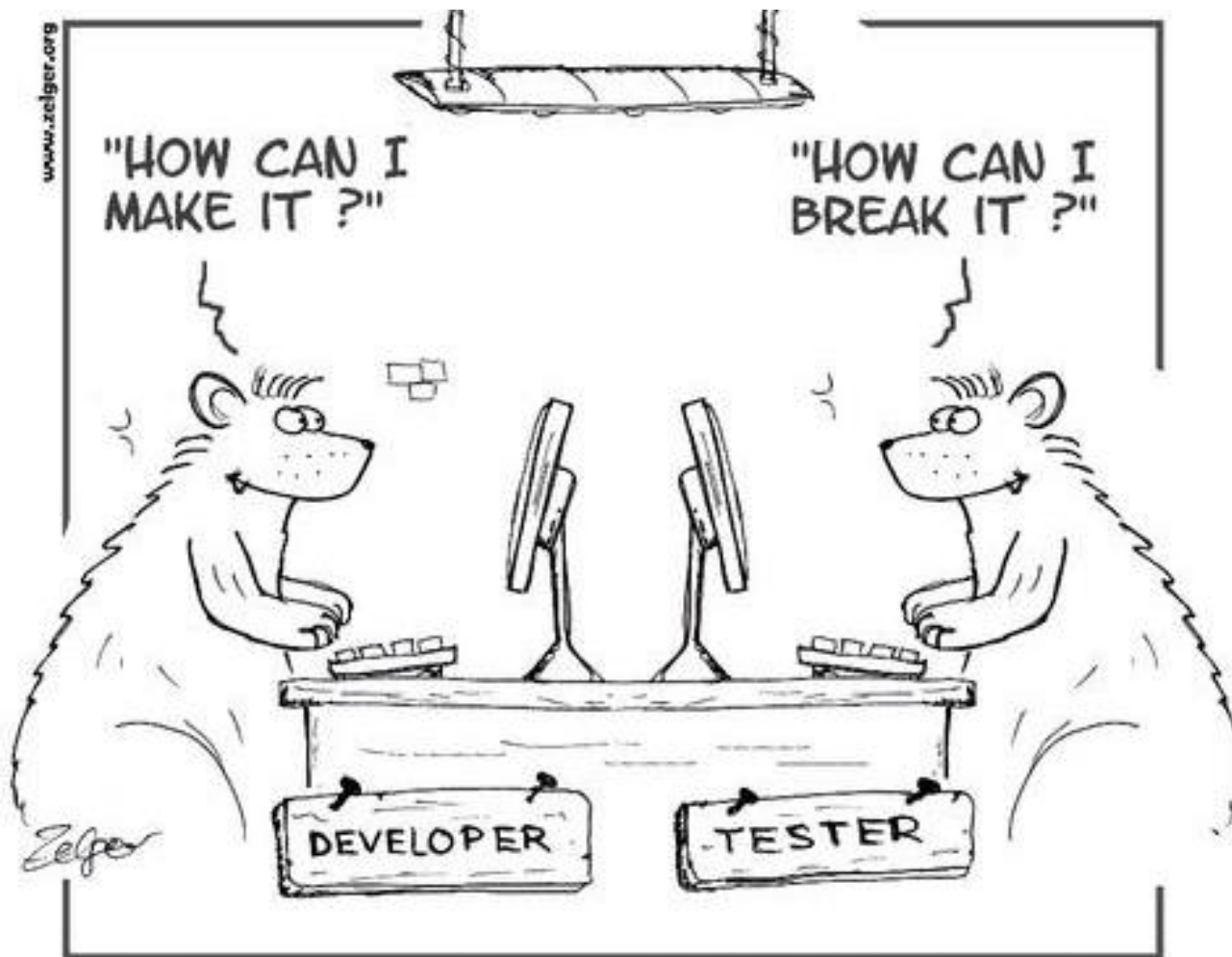
# 軟體測試重要性

## 修正軟體錯誤的相對成本



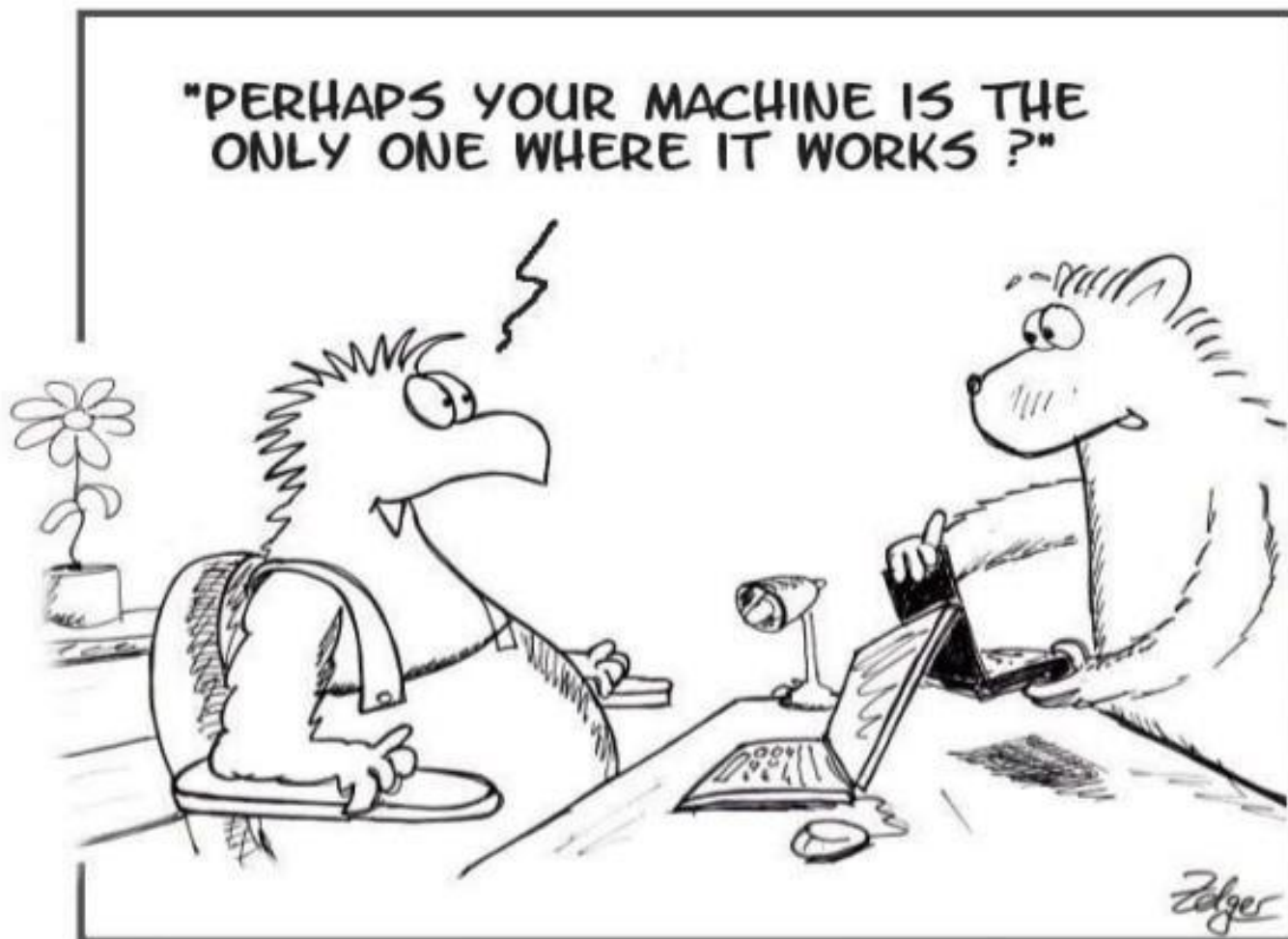


# Software Developer vs. Software Tester





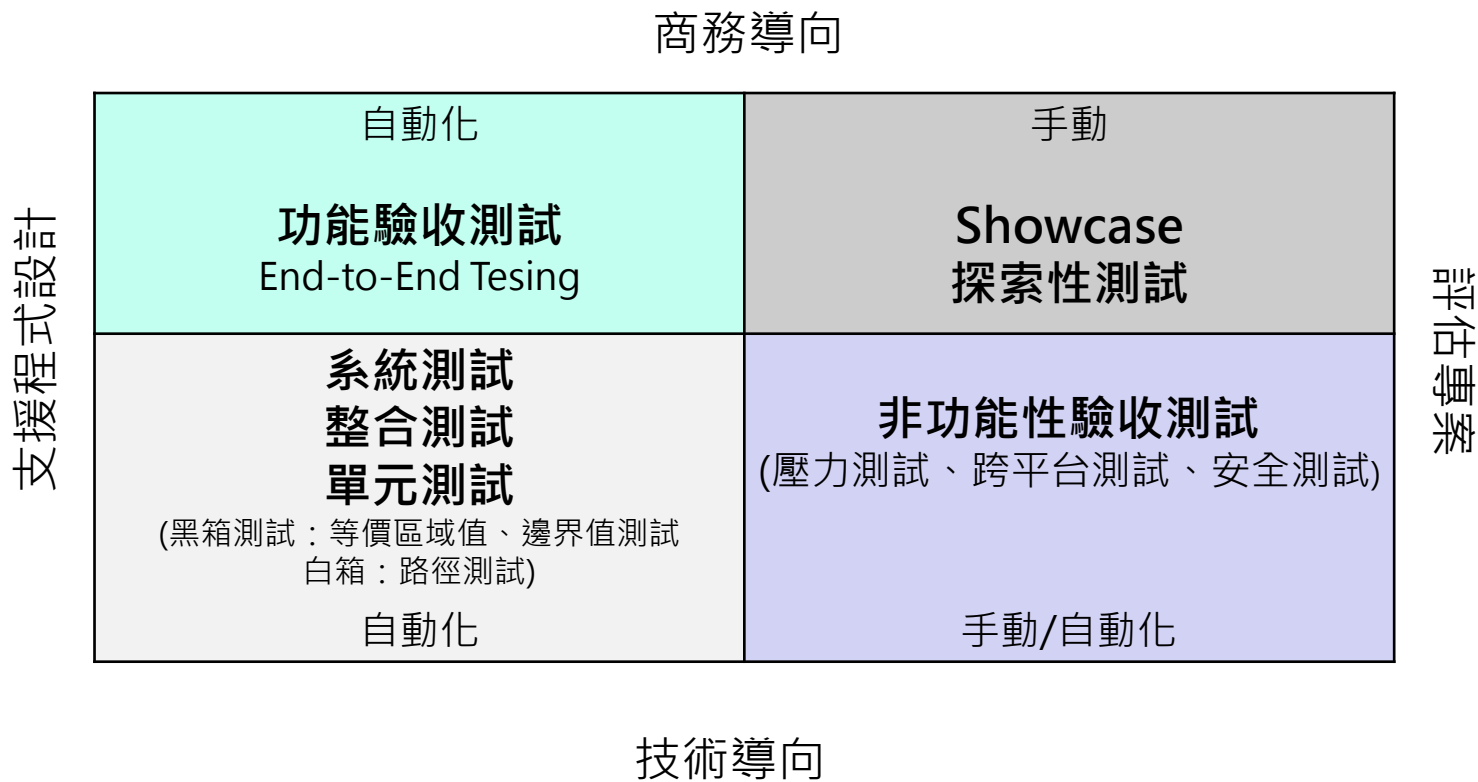
# Why needing an independent tester



It works on my machine



# 軟體測試類型







# My system is well tested!?

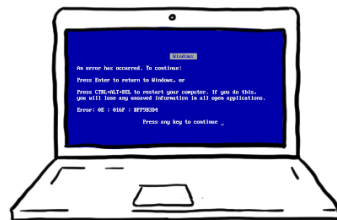
When the developer tests



When the quality team tests



When the project manager tests

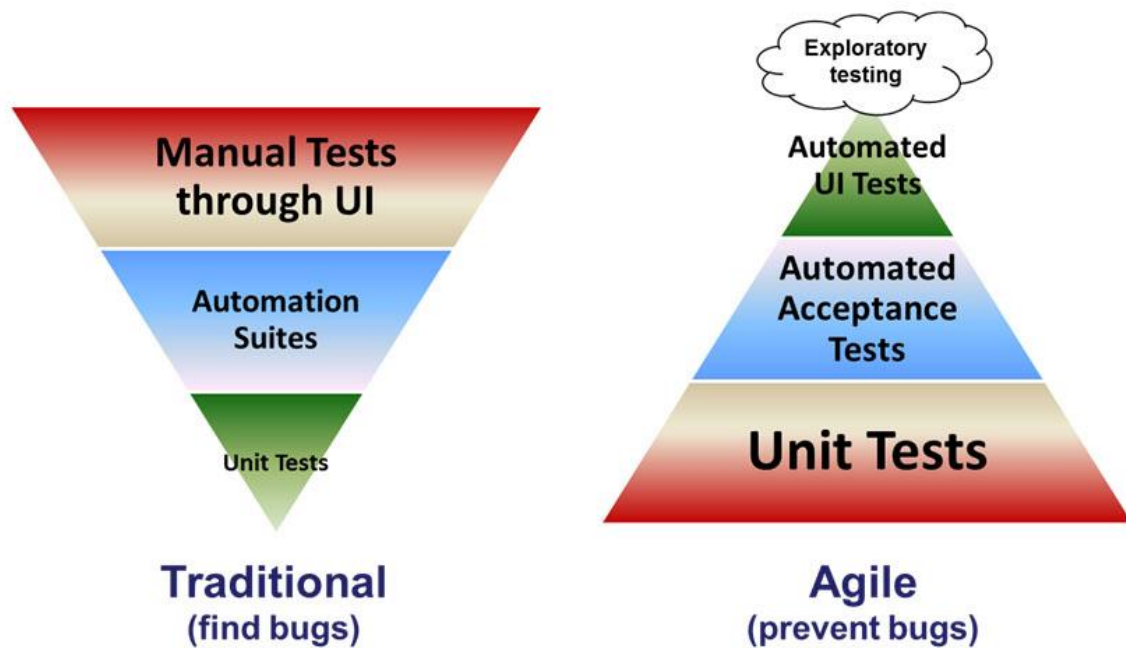


When the customer tests





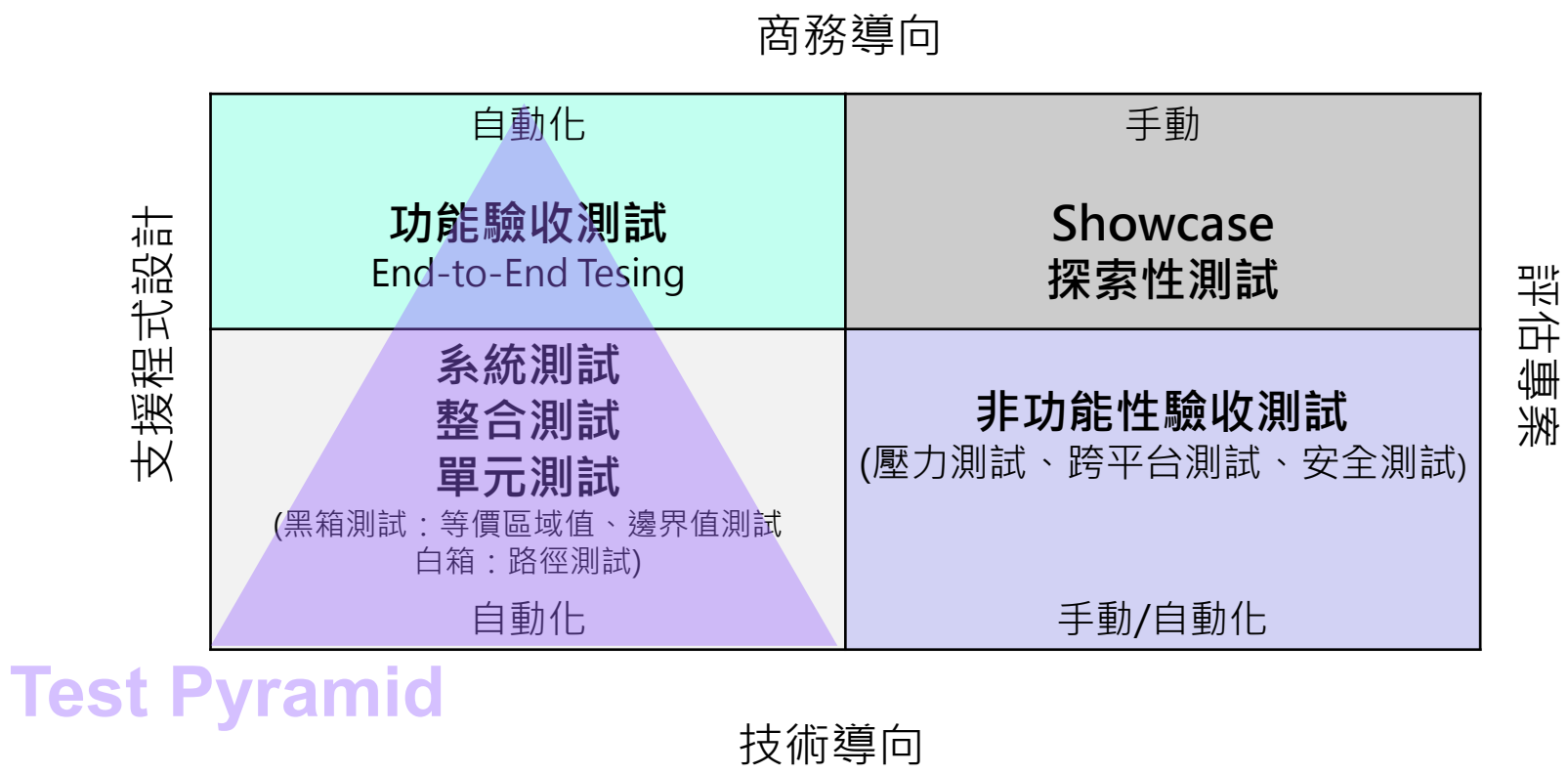
# Testing Effort



[2014] Roger Brown [The Agile Testing Pyramid](#)

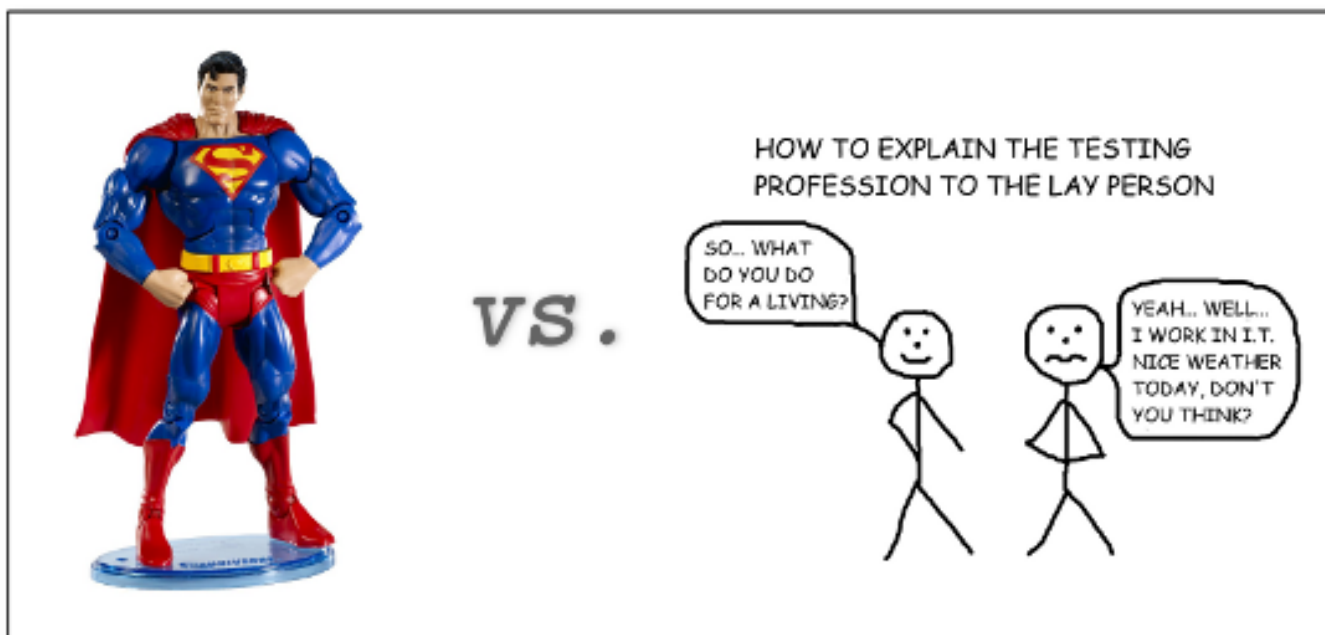


# Test Pyramid





# Coding Heroics vs Testing Process

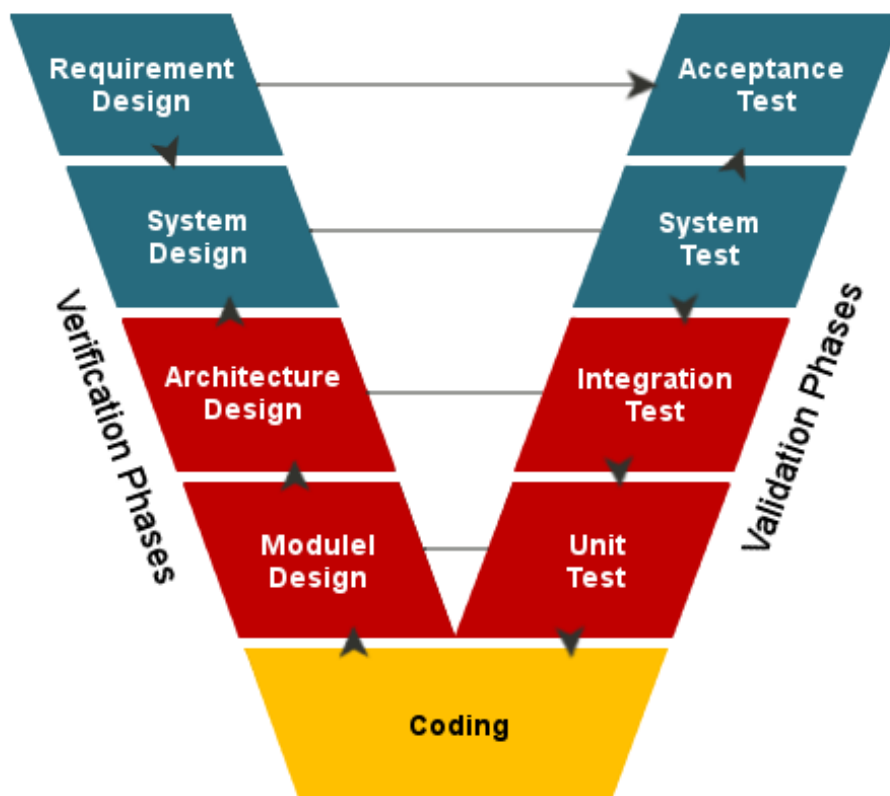


- How risk averse your company/team is
- Stability requirements of your product
- Size and make up of the team

<https://trilemma.technology/2012/02/01/coding-heroics-vs-testin-delivering-software-efficiently-and-effectively/>



# Testing Phases

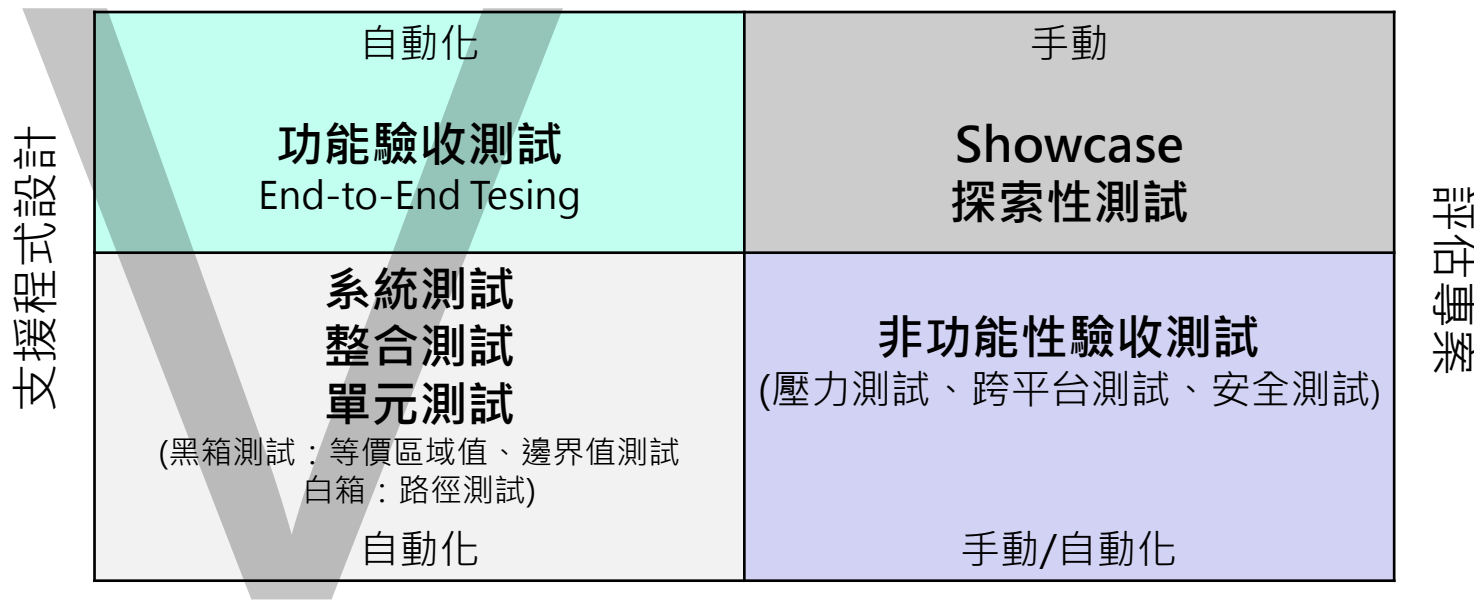




# V-Model

V-Model

商務導向



技術導向



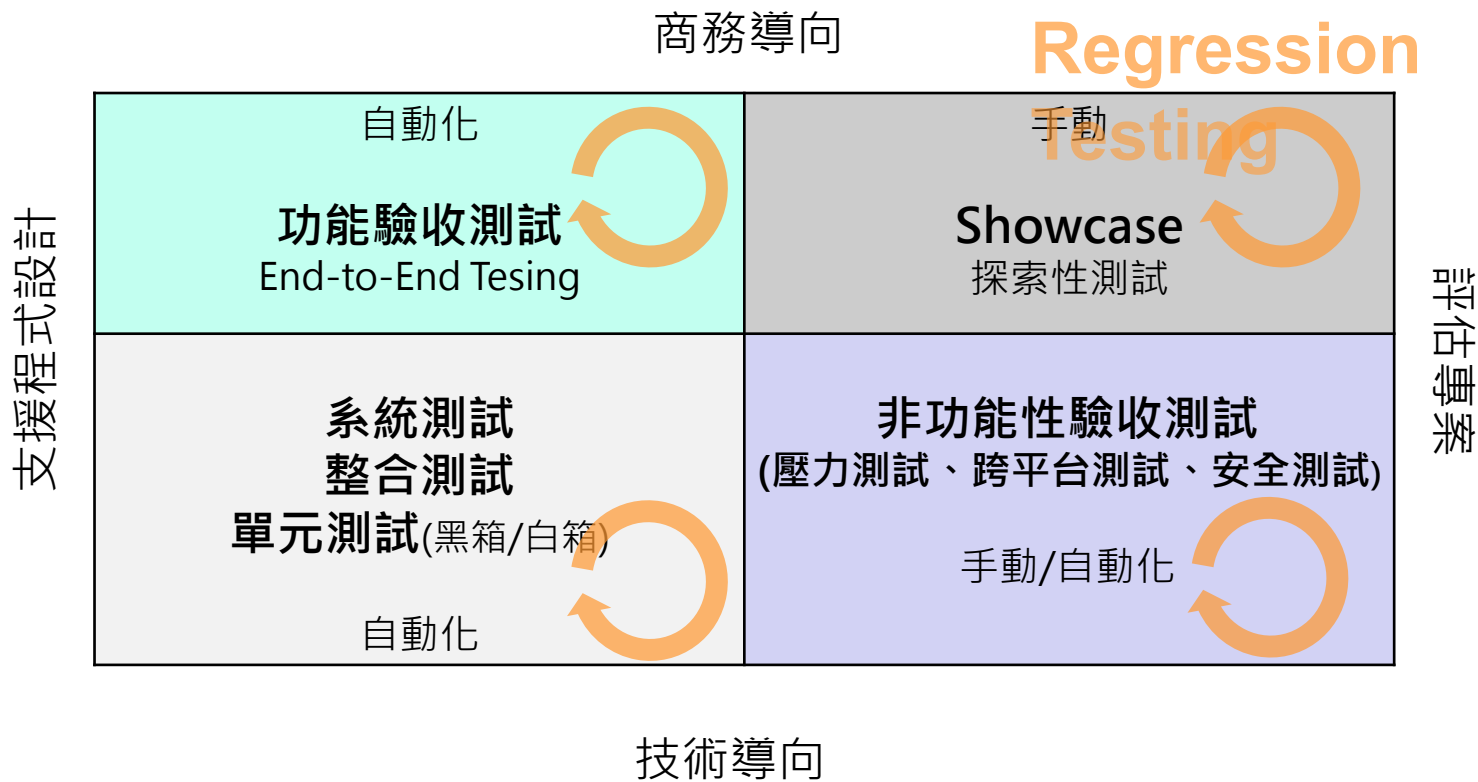
# Why regression testing?

Regression:  
"when you fix one bug, you  
introduce several newer bugs."





# Regression Testing (回歸測試)





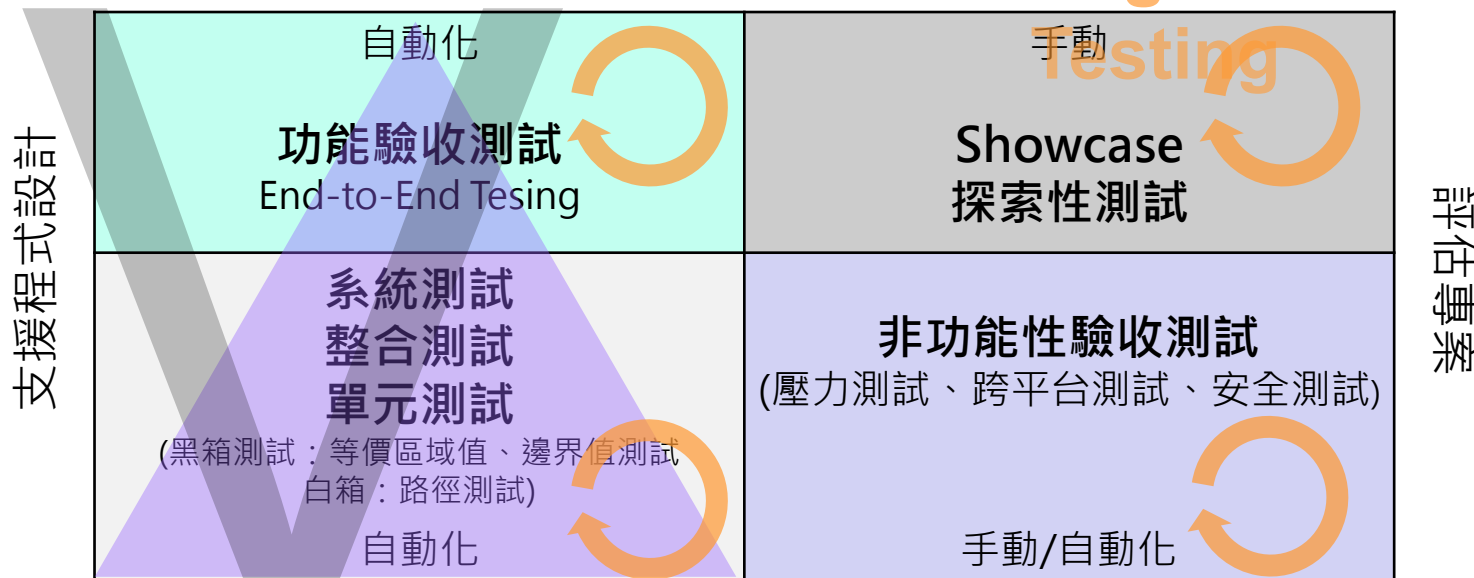


# 軟體測試地圖

V-Model

商務導向

Regression



Test Pyramid

技術導向



# 動態測試案例設計

## □ 測試案例(Test case)

- 內容應包含該測試案例的編號、名稱、目標、條件、輸入資料、測試步驟及預期的輸出結果
- 儘可能在系統發展早期即設計測試案例

## □ 由於單一測試案例無法達成測試目的，測試團隊通常會針對軟體系統設計大量的測試案例，這些測試案例所形成的集合即稱為測試套件(Test Suite)



# 測試案例的設計原則

- 可偵測出錯誤的機率高
- 同類型中最好或較好的
- 複雜度適中
- 不要有重複出現
- 可以清楚的呈現出軟體的錯誤種類



## Lab 6-1

- 請依據假想系統的需求，撰寫出多項測試案例
  - 測試案例編號
  - 名稱
  - 目標
  - 條件
  - 輸入資料
  - 測試步驟
  - 預期的輸出結果

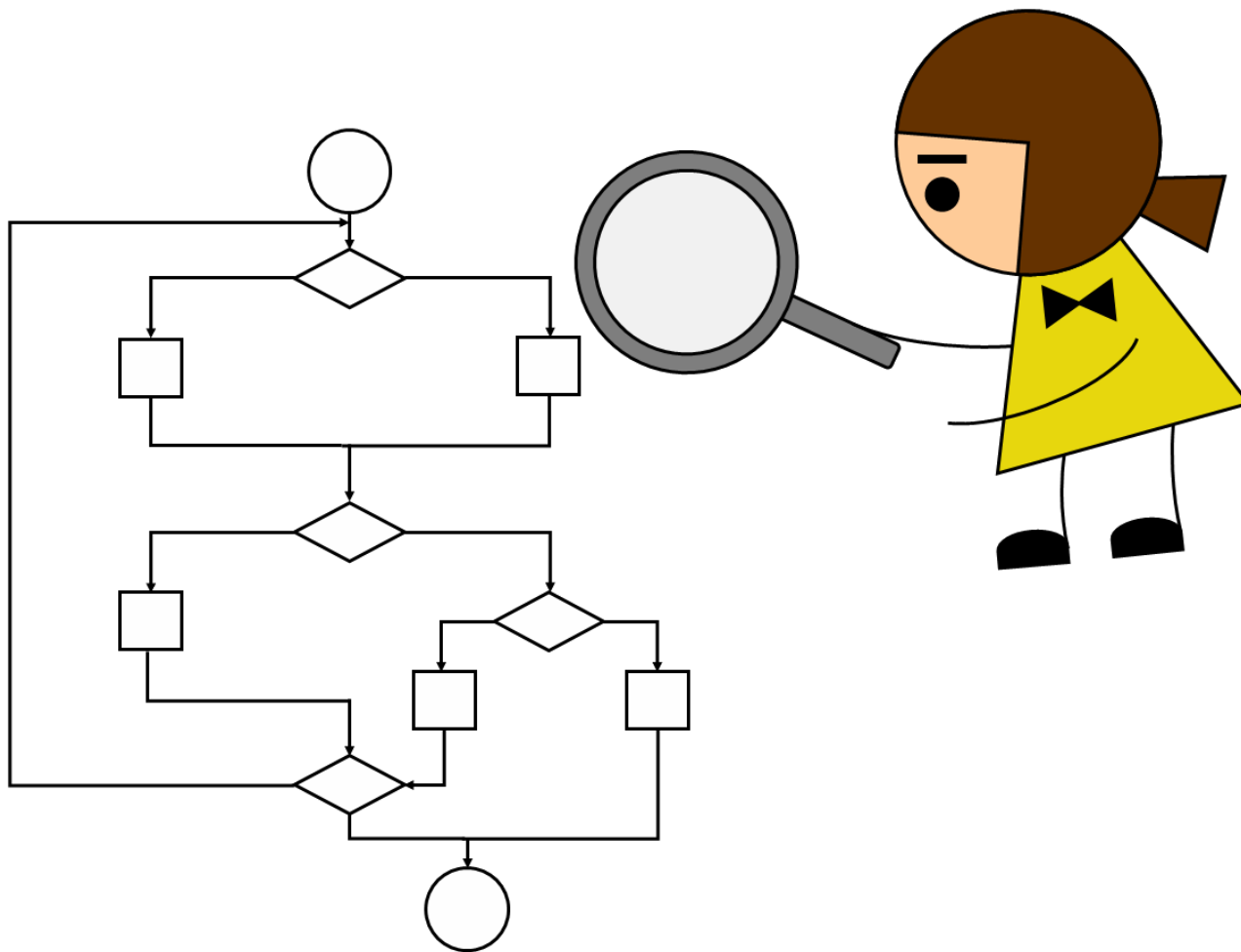


# 白箱測試(White-box testing)

- 又稱為結構測試(Structural testing)
- 了解軟體內部結構後進行測試
  - 根據程式碼判斷條件或迴圈，定義邏輯路徑，產生測試案例資料
- 有效測試程式細節，確保程式的穩定性
  - 對程式結構或資料變數使用的錯誤特別有效
- 缺點
  - 成本高
  - 對需求規格錯誤的效果不好
  - 對介面錯誤的效果不好



# 白箱測試(White-box testing)



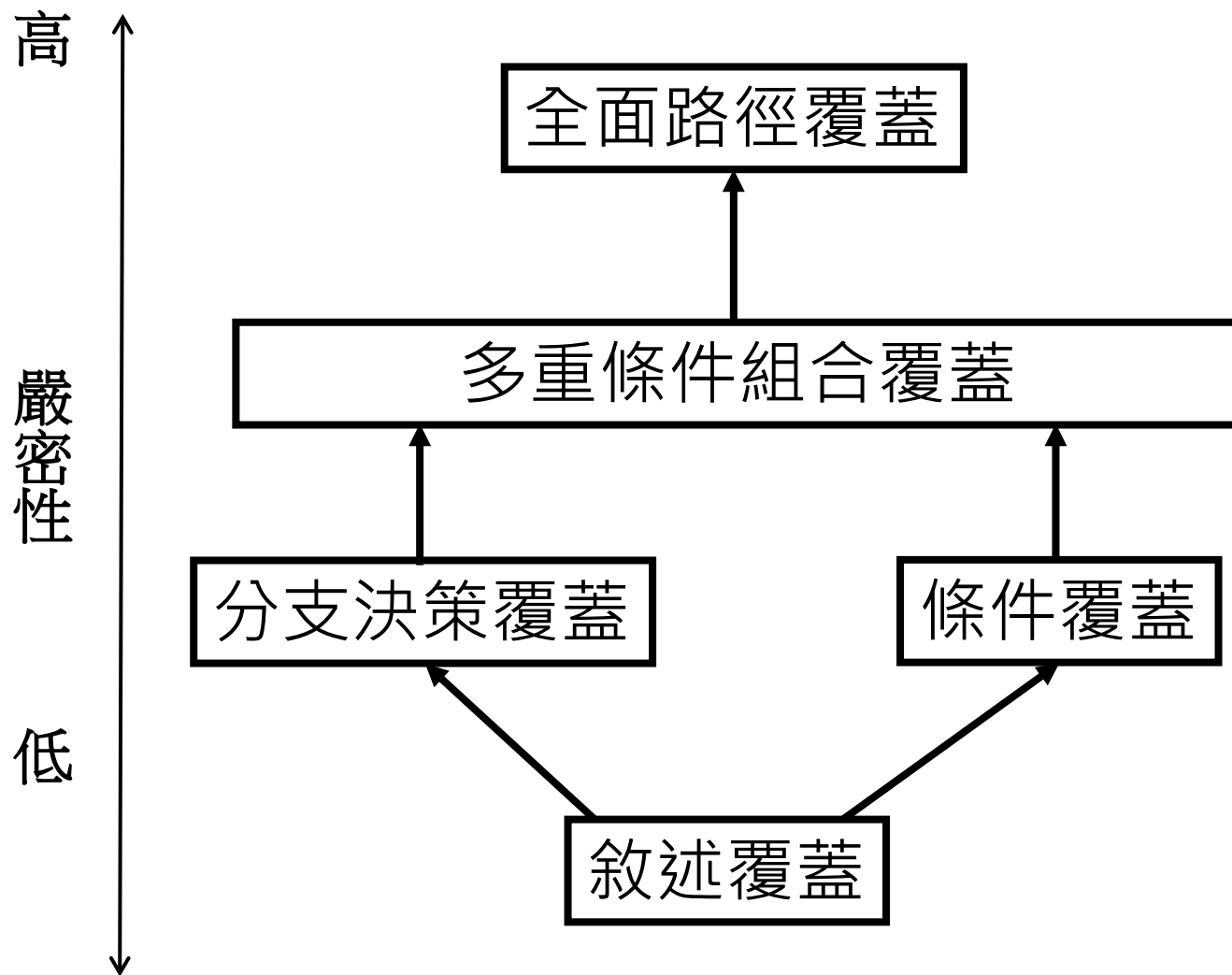


# 測試覆蓋準則 Coverage Criteria

- ❑ 敘述覆蓋(Statements Coverage)
  - 測試案例要使程式每一條敘述至少執行一次
- ❑ 分支決策覆蓋(Branches/Decision Coverage)
  - 測試案例要使程式每一個決策點至少執行一次
- ❑ 條件覆蓋(Condition Coverage)
  - 測試案例要使所有的邏輯判斷情況都至少執行過一次
- ❑ 多重條件組合覆蓋(Multiple-condition Combination Coverage)
  - 測試案例要使不同組合的判斷情況都至少被執行一次
- ❑ 全面路徑覆蓋(All-Paths Coverage)
  - 測試案例要使軟體每一路徑至少執行一次
- ❑ 後續章節將以「shootingGame4SEP」中的程式碼片段搭配數段簡易的程式碼說明各種測試覆蓋準則



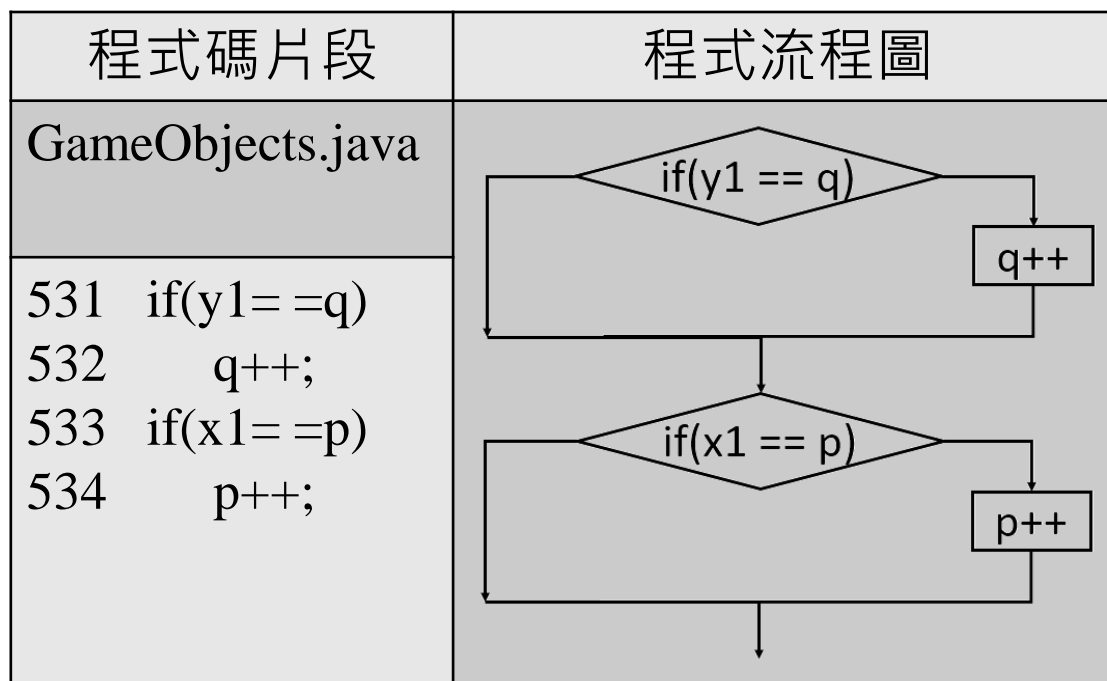
# 測試覆蓋率層次







# 敘述覆蓋Statement Coverage



- ❑ 設計測試案例，使每一條指令敘述至少執行一次
- ❑ 範例1：以上圖中的GameObjects.java程式碼片段為例，說明敘述覆蓋標準



# 敘述覆蓋Statement Coverage

- ◎ 在範例1中，若  $y1 == q$  且  $x1 == p$  時，即可覆蓋所有可執行的指令；結果會使  $q$  和  $p$  的值皆被加1
- 敘述覆蓋測試案例設計方法較不嚴謹，例如程式若包含以下錯誤，程式執行結果不變，所以找不出這種錯誤
  - 行號531的  $==$  誤寫成  $>=$
  - 行號533的  $==$  誤寫成  $<=$
- 敘述覆蓋是最弱的邏輯覆蓋準則，白箱測試至少要做到此測試



## Lab 6-2

---

- 請依據過去課程中撰寫的程式，條列出多項白箱測試案例的Input值

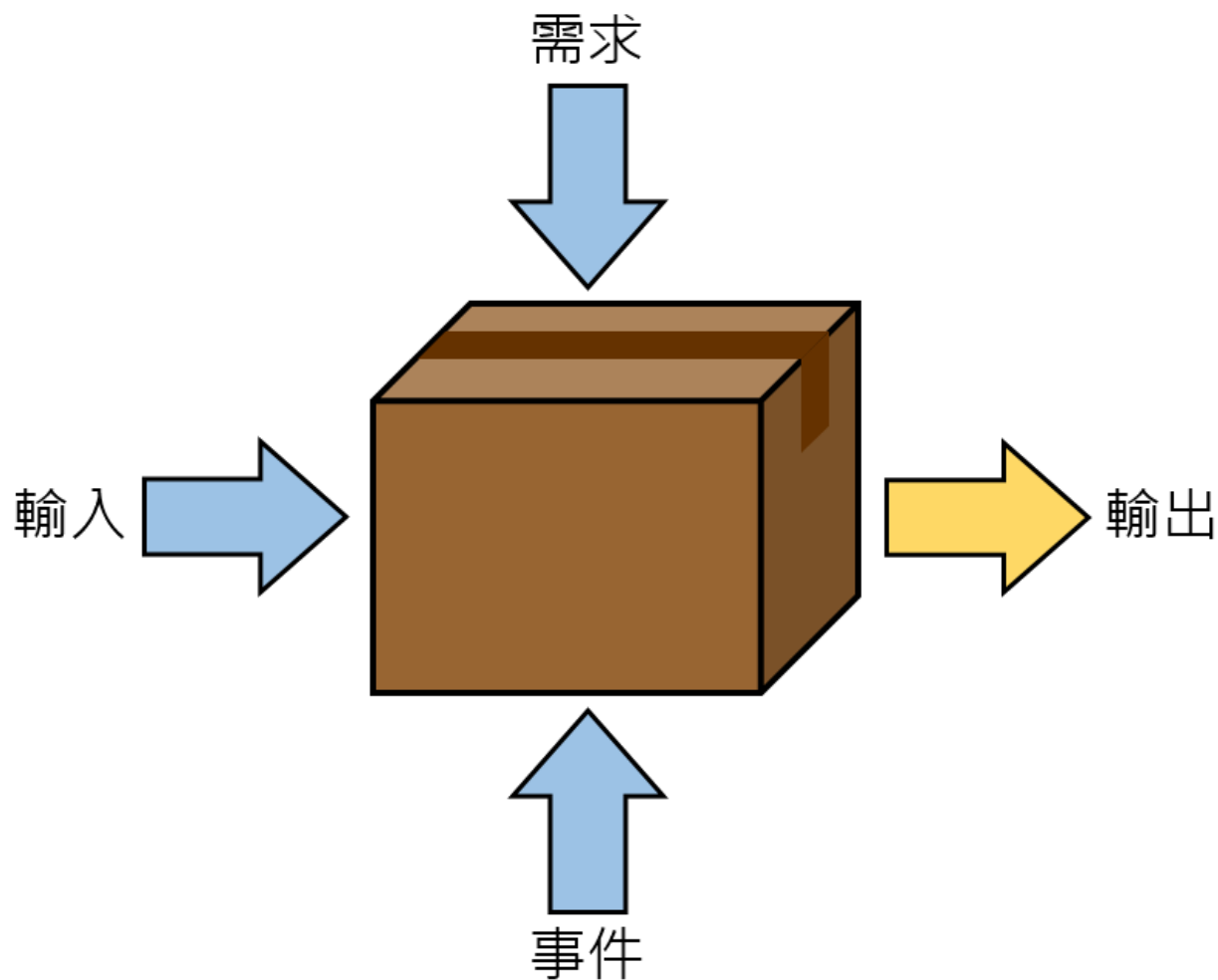


# 黑箱測試(Black-box testing)

- 又稱為功能性測試(Functional testing)
- 了解軟體產品需求功能後進行測試
- 不考慮軟體內部邏輯的結構
- 根據軟體規格書或軟體模組界面，檢驗每組輸入值  
在執行後所產生的輸出值是否符合預期的輸出值



# 黑箱測試(Black-box testing)

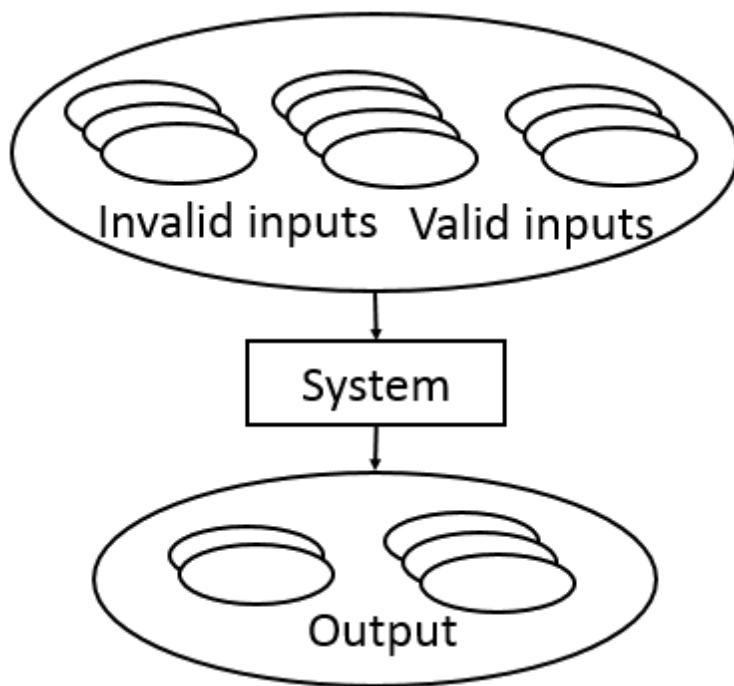




# 等價劃分

□ 以不同資料類別、劃分規則，切割輸入域，設計測試案例

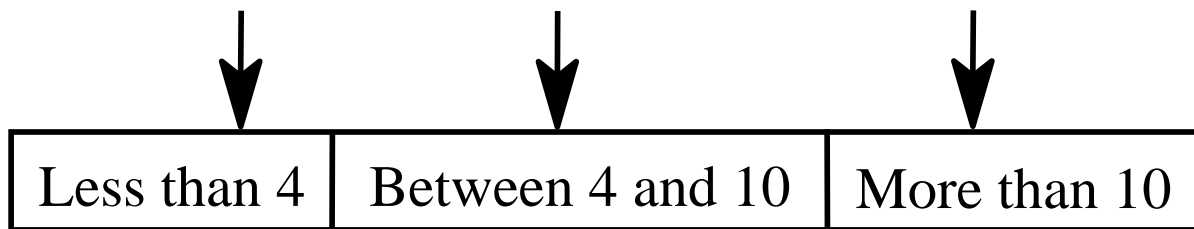
- 輸入域的子集合-各輸入資料對程式中的錯誤都是等效的
- 同一類測試案例，會發現某類同樣錯誤，如所有字元資料的錯誤處理





# 等價劃分原則<sub>(1)</sub>

- 輸入條件規定範圍的a和b值，則取一個有效等價類別 $a < x < b$ ，和兩個無效等價類別 $< a, > b$ 的值設計測試案例
  - 輸入x為， $4 < x < 10$
  - 選擇測試案例為邊界值：3,7,11
  - 選擇測試案例為：負數、非數字





## 等價劃分原則<sub>(2)</sub>

- 輸入條件規定值的個數 $N$ ，則取一個有效等價類別 $N$ ，和兩個無效等價類別 $>N$ ,  $<N$ 的值設計測試案例資料
  - 輸入班級所有人成績，計算全班平均，班級人數為 $10 \leq N \leq 75$ 人
  - 輸入9, 60, 76人的成績





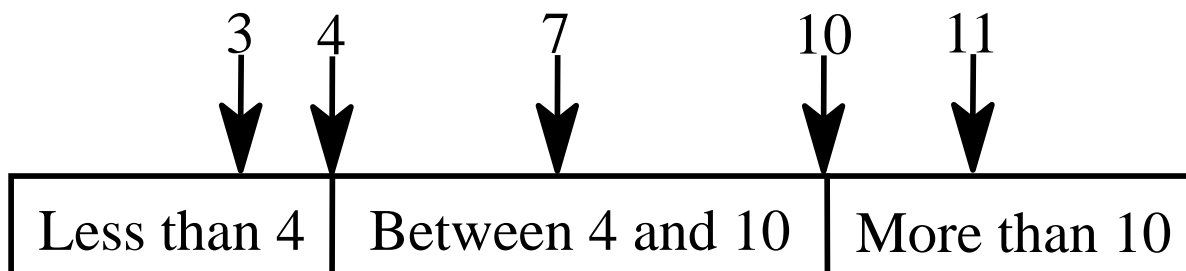
# 邊界值分析

## □ 邊界值分析(Boundary Value Analysis，簡稱BVA)

- 錯誤發生在輸入域邊界上的機率遠大於輸入域的中間

## □ 輸入條件規定值的範圍a和b，則取剛好小於、等於a，和剛好大於、等於b的值設計測試案例

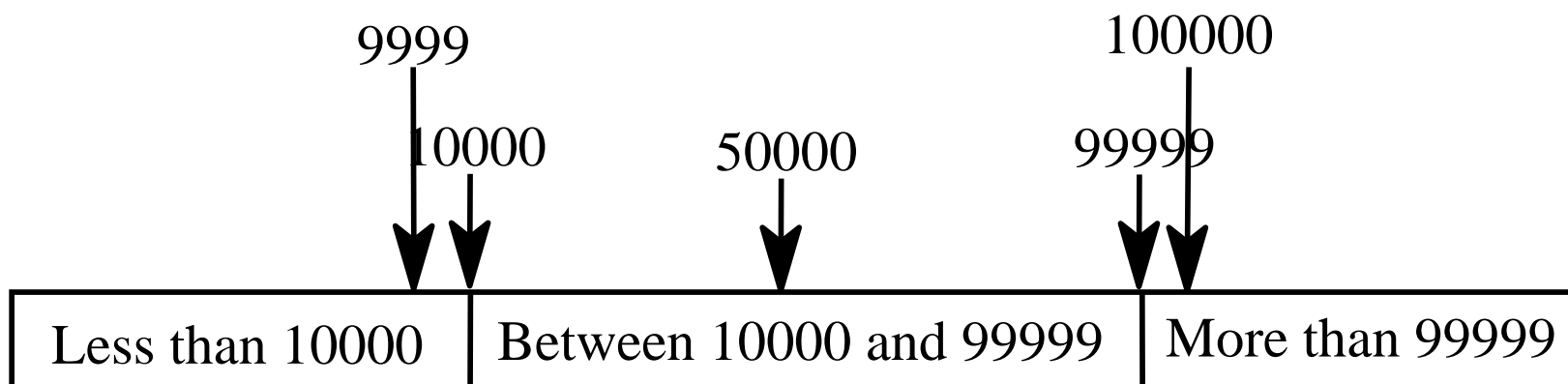
- 輸入x為， $4 < x < 10$
- 選擇測試案例為邊界值：3, 4, 7, 10, 11





# 邊界值分析

- 輸入條件規定一個數值，如整數，則取剛好大於最小值及剛好小於最大值
  - 輸入x為5位整數， $10000 \leq x \leq 99999$
  - 選擇測試案例為邊界值：9999, 10000, 50000, 99999, 100000





# 腦力激盪

- 輸入班級所有人成績，計算全班平均，班級人數為10~75人
  - 請問可測試輸入？人的成績



## Lab 6-3

---

- 依據假想系統的測試案例，條列出多項黑箱測試案例的Input值