

初級行動裝置程式設計師

軟體測試與除錯

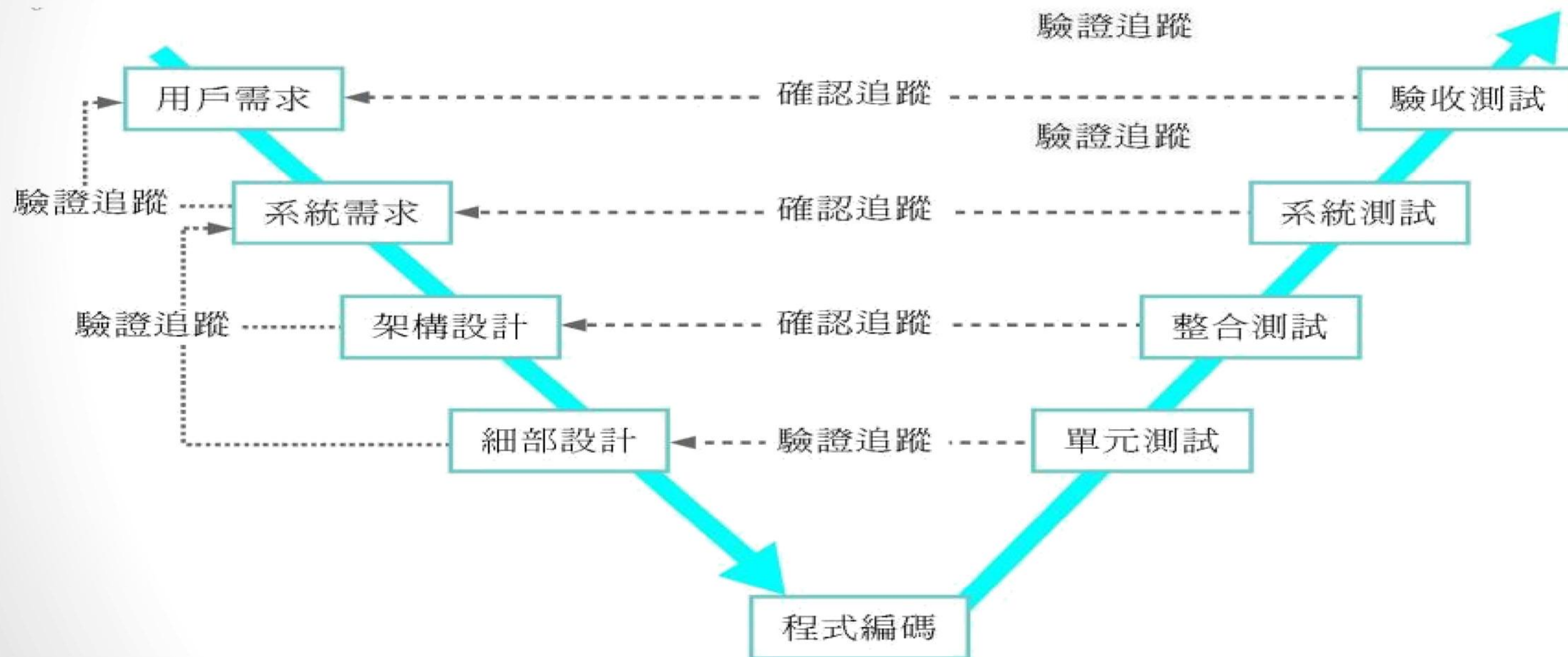


軟體測試的定義與目的

◆ 越早發現軟體問題，開發費用越低

- 撰寫程式後修改軟體錯誤的成本是撰寫程式前的10倍。
- 產品交付後修改軟體錯誤的成本是交付前的10倍。
- 軟體品質越高，軟體發行後的維護費用越低。
- 軟體開始規畫時即應考慮測試的規畫時程及人力：
 - 測試花費，占整個軟體發展工作至少30%以上時間及成本。
 - 需求分析與設計發生錯誤約占65%，程式撰寫約占35%。
- 早期觀點：Analysis → Design → Building → Testing
- 現在觀點：Analysis、Design、Building、Maintenance，全程Testing

軟體測試V模型



軟體測試方法

◆ 靜態分析(Static Analysis)

- 不直接執行受測軟體，以人工或自動化方法評估軟體是否滿足規範。
- 分為審查與檢視方法。
- 規範技術複審、文件複審、資料庫複審、演算法分析審查

◆ 動態測試

- 程式開發後進行，執行軟體程式碼。
- 分為黑箱測試與白箱測試。
- 功能測試、性能測試
- 檢查軟體界面及內部機制正確性。
- 須從系統非預期的行為、或錯誤結果推斷錯誤程式碼。
- 必須花許多時間診斷。

靜態分析優點

◆ 降低專案整體成本。

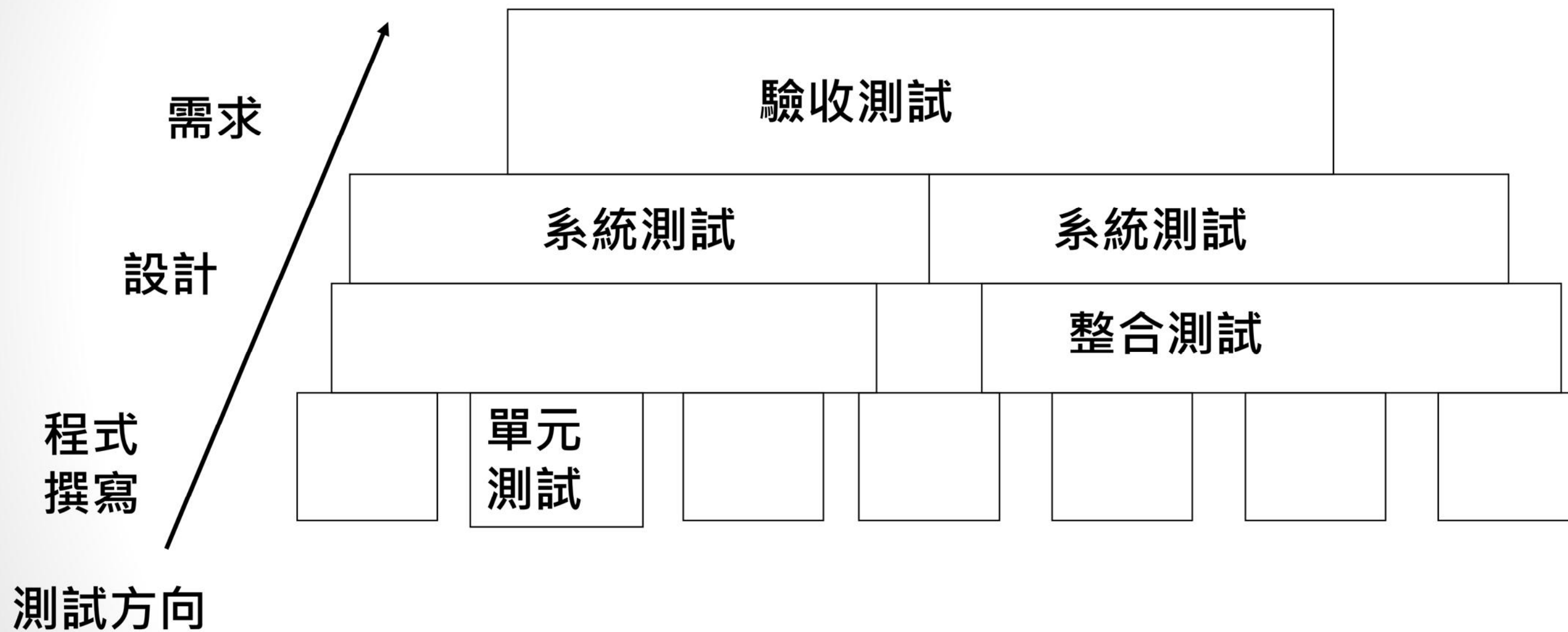
- 超過一半的軟體錯誤發生在需求或設計規格錯誤，審查與檢視可以在程式撰寫前發現錯誤，早期發現問題，可降低除錯與重測成本。

Defect distribution		Relative cost to fix
Requirements	57%	X
Design	27%	X*10
Coding	7%	X*100
Others	9%	X*1000

◆ 靜態分析比動態測試有效率

- 清楚發現錯誤所在及原因，不需經過繁複鑑定錯誤、分析原因、修改與重測，可迅速修正錯誤。
- 有效找出規格模糊、程式撰寫風格錯誤、不合規定程式、與無效程式段落等動態測試難以發現之問題。
- 審查與檢視透過小組討論可以交換成員觀念與想法，協助經驗傳承與訓練溝通。

軟體動態測試程序



軟體動態測試程序

◆ 測試階段

1. 單元測試
2. 整合測試
3. 系統測試
4. 驗收測試

◆ 各階段由不同人員及方法執行，以避免盲點。

◆ 各階段的工作執行時機各不相同，應以時程安排及人力作適度的調整。

單元測試 (unit test)

- ◆ 針對類別中的細小個體去做測試。
- ◆ 從系統中抽離個別模組出來進行測試。
- ◆ 低階測試
- ◆ 獨立測試
- ◆ 細節容易被看清楚
- ◆ 通常由程式設計師自行測試
- ◆ 又稱為元件測試、模組測試、程式測試

整合測試 (integration testing)

- ◆ 穩定度與功能性上的測試，整合了內外部與所支援的額外軟體。
- ◆ 結合一些模組來進行測試，以確定彼此的介面是否運作正確。
- ◆ 多個單元測試
- ◆ 可用來測試無法單一測試的群組
- ◆ 單元間的溝通
- ◆ 非功能性觀點
- ◆ 測試策略
 - Top-down
 - Bottom-up
 - Functional

系統測試 (system testing)

- ◆ 對軟體的整體進行測試。確認整個系統是否符合規格和需求
- ◆ 系統測試是最後的整合步驟

驗收測試 (acceptance testing)

- ◆ 以專案客戶制定的驗收標準去測試產品。
- ◆ 確認所被實現的功能是否符合當初使用者所要求，解決使用者想解決的事。

動態測試方法

- ◆ 程式開發後進行，執行軟體程式碼。
- ◆ 功能測試、性能測試。
- ◆ 檢查軟體界面及內部機制正確性。
- ◆ 以下二種測試方式：
 - 黑箱測試
 - 只管輸入與輸出，主要以產品的功能為指標的測試方法。
 - 白箱測試
 - 測試人員直接在軟體程式上進行測試，測試包含了語法、邏輯、路徑等測試。

動態測試方法

- ◆ 測試階段的工作是檢測在製作階段完成之成品的正確性
- ◆ 測試階段（testing phase）的目標就是找出錯誤。
- ◆ 測試的兩種基本方法：
 - 白箱測試（white box testing）
 - 以程式本身內容為導向
 - 設計一些測試案例，以強迫執行模組中的某些敘述或某些順序的敘述
 - 黑箱測試（black box testing）
 - 以程式功能為導向（不理會程式內容如何撰寫）
 - 「測試案例」（test case）是依據規格所產生，而不考量程式碼



白箱測試 (White-box testing)

- ◆ 也稱為玻璃箱測試 (glass-box testing)
- ◆ 「白箱測試」是測試軟體元件的內部結構，檢查軟體元件是否能依照設計正確無錯誤的執行
- ◆ 測試者了解受測軟體的內部結構，並在設計測試時利用這個知識。
- ◆ 測試的目標就是檢查軟體所有組成元件，是否符合所設計的要求
- ◆ 基本路徑測試與控制結構測試是其中兩種測試方法。

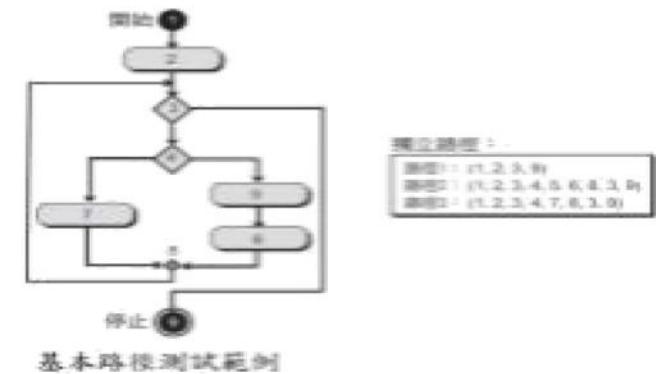
- **基本路徑測試** (basis path testing) :

會產生一組測試數據，可以讓軟體的每個敘述最少執行一次。

- **控制結構測試** (control structure testing)

所包含的測試較基本路徑測試要廣泛，此法使用數種不同的測試種類。

- 條件測試
- 資料流程測試
- 迴圈測試



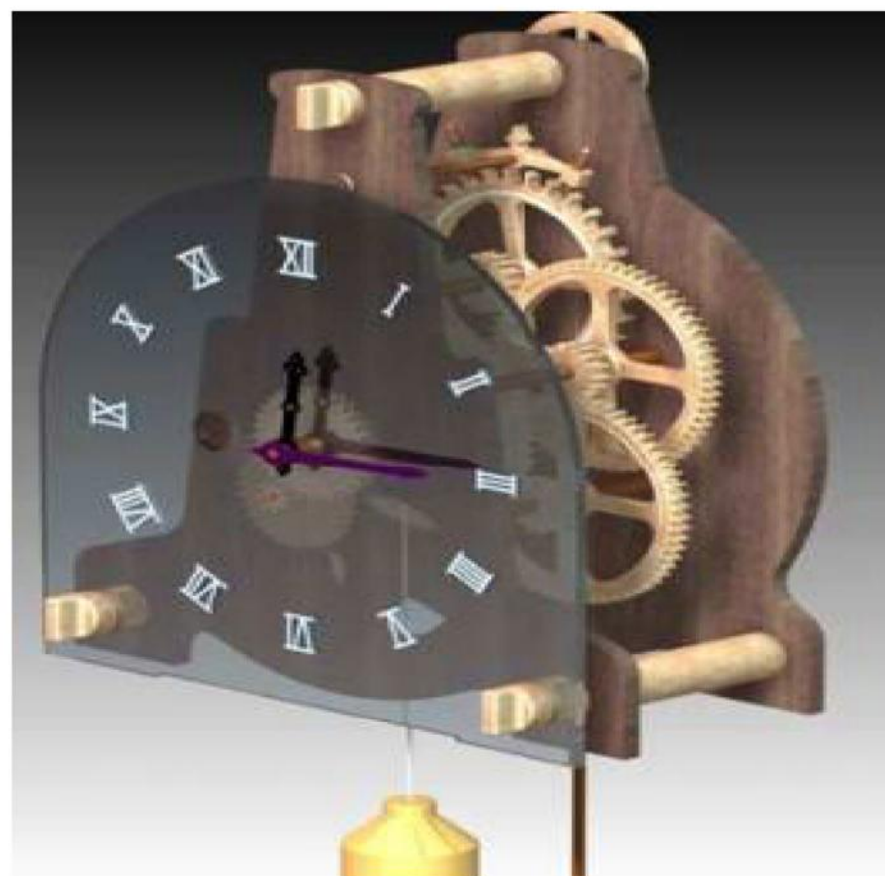
白箱測試 (White-box testing)

- ◆ 本技術的測試實例是以程式的控制結構(control structure)為依據進而設計而得，包括的測試細節如下
 - 測試所有內部資料結構
 - 對於所有含有「條件結構」及「反覆結構」中的布林運算式之所有可能值均必須被執行過
 - 對於所有「反覆結構」的內部敘述與所有條件可能值均須檢查
 - 確保程式中的每一組「獨立路徑」控制流程均應被檢查

動態測試案例設計

◆ 白箱測試(White-box testing)

測試輸入



執行結果
與期望結果比對

黑箱測試 (Black-box testing)

- ◆ 命名的由來是因為所測試的軟體內容未知，也不知道它是如何運作的。
- ◆ 黑箱測試是依據軟體元件由外部所見的功能來測試軟體元件是否能夠正確執行
- ◆ 此類測試主要的目標是以程式功能為導向
- ◆ 通常「白箱測試」會先做，確認程式本身沒問題後才會執行「黑箱測試」
- ◆ 測試過程應特別注意檢驗輸入資料的最大可能值及最小可能值之執行結果是否有誤
- ◆ 不依賴對軟體內部結構的認識。純粹以使用者角度來看待。
- ◆ 不在意執行細節，只在意工作效率。
- 徹底測試
- 隨機測試
- 邊界值測試

動態測試案例設計

◆ 黑箱測試(Black-box 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是否正確
- 測試軟體界面，正確輸入、輸出

測試輸入



執行結果
與期望結果比對

動態測試案例設計

- ◆ 假設有一個程式可針對最多50筆輸入的整數資料(值介於0~100之間)進行排序，輸出為排序之結果
 - 執行受測程式內的每一個敘述至少一次，以確定程式沒有錯誤，這類的測試模式屬於「白箱測試」
 - 若以最大值(100)和最小值(0)，以及最靠近最大值(99)和最小值(1)的數值來做為輸入資料，這種測試方法稱為邊界值分析法(boundary testing)，屬於「黑箱測試」的一種

α 測試 / β 測試 / γ 測試

- ◆ α 、 β 、 γ 常用來表示軟體測試過程中的三個階段，基本上，順序是由 α 測試開始，之後到 β 測試，以及 γ 測試。
 - ◆ α (Alpha) 是第一階段，一般只供內部測試使用；
 - ◆ β (Beta) 是第二階段，已經消除了軟體中大部分的不完善之處；
 - ◆ γ (Gamma) 是第三個階段，此時產品已經相當成熟，只需在個別地方再做進一步的優化處理即可上市發行。
- ◆ α 測試、 β 測試、 γ 測試雖然只是根據不同時間所有不同的測試名稱，不過由於在不同階段產品所著重的關注點不盡相同，因此需要使用的測試技巧也會有所不同。
 - ◆ α 測試僅開放給內部測試，可說是在測試此軟體的骨架，在虛擬或是套好的資料上作測試以期掃出嚴重大問題。
 - ◆ β 測試有時會開放給由外部人員參與測試，在真實的環境下使用實際的資料做測試，確認系統、效能運作正常。
 - ◆ γ 測試是上市前最後的測試，確認重大功能運作正常，該優化的部分都達到標準。

α 測試

- ◆ α 測試通常是產品還在開發中，或是完成部分的時候才開始做測試，可以在程式完成後開始，或是確認產品達到一定的穩定和可靠度之後再開始，通常會利用模擬或是實際操作來做測試，只供給內部測試使用，
- ◆ α 測試可說是在測試此軟體的骨架；直到產品已經做的差不多了，在 α 測試中所找到嚴重的大問題也都修正了，產品也達到一定的可靠度，才可以進入 β 測試。

β 測試

- ◆ β 測試可稱為確認測試，在一個真實的環境中以實際的資料來執行測試，以確認效能、系統執行有效率，系統復原與備份作業正常（註一），透過測試讓資訊系統日後可以更趨完善。
- ◆ β 測試有時會也給由外部人員參與測試，例如有些會將產品讓最終用戶直接實際操作來測試；
- ◆ 我們常在遊戲界聽到某支新遊戲要開放給玩家「封測」或「公測」，此階段也是屬於 β 測試。

註一：在封測過程中，有時候會發生 **server** 崩潰，玩家同時被迫下線，重新登入回來後發現裝備都不見了，原本的進度被拉回一個星期前等等，這就是系統復原和備份作業不正常，β 測試中某些產品會特別注意此段測試，以免日後真正上市發生此問題而造成糾紛。

β 測試

- ◆ α 和 β 測試常見的一個誤解是「 β 測試 = 黑箱測試」。
- ◆ α 和 β 測試指的是根據產品的階段來對應是什麼測試，例如 α 測試就是對應到 α 階段產出的產品，對此產品做測試，而 β 測試則是對應到 β 階段產出的產品做測試；
- ◆ 而白箱、黑箱和灰箱測試技術是從技術和方法層面對測試的描述。
- ◆ 兩個概念看的角度不相同，不應該將這兩部分概念混淆。

Q & A