

# 同仁審查之流程與關鍵因素探討

## —以台灣證券交易所系統設計審查為例

劉臺雄

證券櫃檯買賣中心

Tel: 886-2-2366-6020, Fax: 886-2-2369-1223

thliu@mail.otc.org.tw

時序時\* 楊維楨

淡江大學管理科學研究所

Tel: 886-2-8631-3221, Fax: 886-2-8631-3214

hshih@mail.tku.edu.tw; 018467@mail.tku.edu.tw

周清江 徐煥智<sup>#</sup>

淡江大學資訊管理學系

Tel: (886) 2-2621-5656 ext. 2846; Fax: (886) 2-2620-9737; cjou@mail.tku.edu.tw

#Tel: (886) 2-2621-5656 ext. 2881; Fax: (886) 2-2620-9737; shyur@mail.im.tku.edu.tw

### 摘要

近年來政府為提升國內軟體開發組織／公司的競爭能力，並爭取國際代工機會，開始推展軟體的能力成熟度整合模式 (capability maturity model integration, CMMI) 認證制度。CMMI共分為5級，在第三級 (Level 3) 流程中，必須完成軟體驗證 (verification) 之「同仁審查 (peer reviews)」特定目標。軟體的同仁審查方法國外已研究多年，但依最正式的方法執行，會耗費相當大的人力及時間成本，對國內大多數的中小型軟體開發組織幾乎無法執行。如採非正式的方法，又則無法達到最高效率，因此本研究探討同仁審查活動的關鍵因素，分析各因素之重要性順序；並以台灣證券交易所同仁審查流程為個案研討對象，期望研究結果可做為未來執行非正式同仁審查之參考。

本研究先探討同仁審查重要成功關鍵因素 (critical success factors, CSF)，再使用「德菲法 (Delphi technique)」問卷，請相關專家學者確認並評估關鍵因素重要性。其次以「分析層級程序法 (analytic hierarchy process, AHP)」分析專家回應資料，根據分析結果提出個案流程改善意見。最後再另請不同專家小組就前述建議事項，評估其可行性及執行績效。根據專家評估認為同仁審查組成要素之重要性，依序為程序、人員、與技術；再依重要性序之前五項關鍵因素，分別是完善的檢查表、規劃的程序、開發流程中應規劃同仁審查流程、審查人員會議前充分的準備、以及設定同仁審查目標等。最後，本研究所建議的流程及技術除有助於台灣證券交易系統之軟體開發外，亦可為其他組織在開發軟體時執行同仁審查之參考依據。

**關鍵詞：**軟體驗證、同仁審查、同仁審查成功關鍵因素、德菲法、分析層級程序法、能力成熟度整合模式

\* -- corresponding author

## 1. 緒論

提升軟體品質及增加軟體製作效率，是軟體開發組織及專家學者持續追求的目標，在「軟體工程」概念下，專家學者業已提出許多改進方法；由同仁以人工審查方法驗證軟體產品，降低軟體由個人製作所產生品質風險，即為「同仁審查(Peer Reviews)」。

同仁審查實施方式呈現多樣化，不同的方式獲得效果不同。本研究探討同仁審查流程之關鍵因素及其重要性，希望研究結果供中小型軟體開發組織參考運用，並達成下列目的：

- (1) 使軟體開發組織瞭解同仁審查活動之成功關鍵因素。
- (2) 提供軟體同仁審查成功關鍵因素的重要性順序（權重）。
- (3) 使同仁審查活動能有效益的被執行。
- (4) 使中小型軟體開發組織之同仁審查活動能依關鍵因素重要性執行。
- (5) 間接的達到軟體組織流程改善目的，加強組織競爭力。

## 2. 文獻探討

在軟體開發或維護過程中，驗證流程是軟體品質保證重要的環節，驗證(Verification)主要功能是於軟體開發生命週期中各個階段(Phase) 作業完成後，對該階段所產出的產品進行驗證，確定其品質是可被接受的 (build the system right)；執行軟體驗證活動是要找出產品留存錯誤，並確定其符合需求。軟體驗證二個最主要的方法是測試(Testing)及同仁審查。所謂同仁(Peer)是指與被審查產品之製作者的同仁，具有相同技術背景，在組織中職位層級相當，沒有長官、沒有主管人員(Chao, 2004)。必須沒有控制其他同事績效考核權力的人(Futrell, 2004)，具備一定水準系統開發技能，並對審查產品知識充分瞭解。審查是軟體製造者及工作同仁對軟體產品有條理的進行驗證，以發現錯誤及需要修改之處。

### 2.1 同仁審查目的

同仁審查經研究證實可以有效去除軟體產品錯誤的機制（資策會，2002）。同仁審查的目的是希望軟體的錯誤能愈早發現愈好，軟體品質控制原則在儘早發現錯誤並立即解決（林信惠等，2005）。在軟體開發專案中因錯誤而導致重複的工作成本幾乎耗費整個專案30-50%成本( Futrell, 2004)。而且軟體在交付顧客後發現錯誤及修復之成本，比需求討論(Requirements)及規劃(Design)階段發現錯誤、修復之成本高100倍，小型或較不重要的系統開發案，也有5倍差距 (Boehm, 2001)。但並非執行同仁審查活動後就可取代測試工作，而是二者相輔相成，方能確實達到軟體品質驗證功能。

### 2.2 同仁審查類別

一般將同仁審查可分為正式 (Formal) 審查與非正式 (Informal) 審查二大類：

#### (1) 正式審查(Formal Review)

正式審查方法依照一套嚴謹且結構化的流程實施審查工作 (Stark, 2003)。是大型軟體開發組織，運用在開發專案或品質要求高的維護專案，正式審查方法已被研究證實比非正式方法有效益，其中檢視 (Inspection) 方法是被認為最正式審查方式，其具備下列特性 (Wieggers, 2002)：

- (i) 定義明確的目的
- (ii) 審查人員均受過審查訓練
- (iii) 會議由受過訓練的主持人掌控
- (iv) 每位參與者被指定擔任固定角色及工作
- (v) 是屬於文件審查的處理流程
- (vi) 訂有明確的允入與允出準則
- (vii) 追蹤所發現的錯誤後續更正情形
- (viii) 製作審查報告做為管理參考數據

## (2) 非正式審查(Informal Review)

非正式審查不需要結構化的流程，不需要正式紀錄文件，也不需要蒐集會議執行的統計資料，完全依照審查單位需求而安排適當的流程。非正式審查與正式審查之區別並無絕對的標準，Wieggers(2002)將審查活動之正式與非正式關係圖示如圖 1。

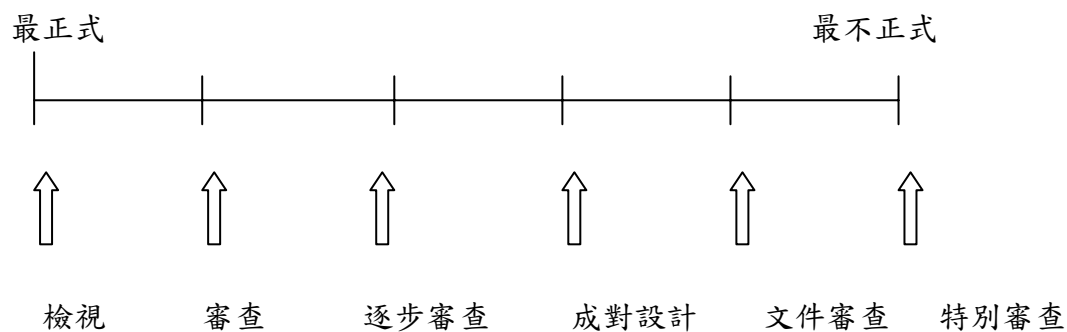


圖 1 同仁審查正式程度表示圖 (Wieggers, 2002)

## 2.3 同仁審查方法

同仁審查活動從正式到非正式有許多方法，專家學者在研究文獻中最常使用檢視(Inspection)、審核(Review)與逐步審查(Walkthrough)三種方法(Futrell, 2004)，分別說明如下：

### 2.3.1 檢視(Inspection)

IEEE 1208 軟體審查標準(1997) 定義「檢視」審查方法是對軟體產品實施視覺式檢驗，以偵測及鑑別軟體的異常事項，包括錯誤，以及對標準及規格書的偏離事項。檢視由經過檢視技術訓練之公正者領導同儕執行檢驗。檢視方法是規劃完善且定義明確的一種群體審查的流程 (Tjahjono, 1996)。

#### (1) 參與人員

檢視審查會議參加人員最好是安排 5-6 人，其參與人之角色有主持人(Moderator)、審查人員(Inspectors)、紀錄(Recorder)、報告人員(Presenter)、製作者(Author)。

#### (2) 審查流程

Fagan (1976)最初提出五步驟的檢視方法，包括準備會議(Overview)、審查前準備(Preparation)、審查會議(Inspection)、修改工作(Rework)、執行追蹤(Follow-up)等程序，經過專家學者不斷研究改進，Wieggers(2002)提出了七大審查流程的方法：

- (i) 規劃(Planning)

軟體開發組織為有效率的執行審查工作，必須周密的規劃活動進度、安排參與人員等工作。

(ii) 預備會議(Overview)

為讓審查者熟悉產品內容，由主持人安排產品作者對受審產品進行說明，並指定活動中審查人員擔任工作角色。

(iii) 準備工作(Preparation)

近年來研究者將焦點從檢視法的群體會議移至準備階段，特別是閱讀方法成為本階段的關鍵(Harjumaa, 2005)。審查人員在此階段必須對所審查產品文件資料進行閱讀，檢查表的製作也由檢查表基礎(Checklist-based)改為情境基礎(Scenario-based)，以協助審查人員發揮最大效力。

(iv) 審查會議(Inspection)

此程序主要是進行審查工作，由報告人分段報告，審查人員提出問題，問題經確認後，由紀錄人員紀錄審查結果。

(v) 修正工作(Rework)

審查會議經確認為產品錯誤之處，產品作者必須確實對產品進行修正。

(vi) 執行追蹤(Follow-up)

為避免會議後對決議事項執行不力，必須追蹤需修正項目是否完成。

(vii) 原因分析(Causal Analysis)

審查活動中所蒐集產品錯誤之資料，最後分析其發生原因，並據以改善開發作業流程，使爾後作業不再發生相同錯誤，同時累積審查經驗。

(3) 檢視審查相關規定

檢視方法要求嚴格遵守審查程序，審查活動還需注意執行技術，以提升效益，經整理其相關規定如下 (Futrell, 2004)：

- (i) 會議進行時間應控制在二小時之內。
- (ii) 為使參加同仁能暢所欲言，負責同仁工作評核之主管不可參加會議。
- (iii) 為追求會議最高效益，會議僅偵測產品問題點，不討論問題解決方法。
- (iv) 產品作者不可擔任報告人，以免引導審查人員錯誤方向。

### 2.3.2 審核 (Reviews)

「審核」審查方法也稱為團隊審查(Team Reviews)或技術審查(Technical Review)，是正式審查的一種 (Stark, 2003)，團隊審查的目的是利用一組具備專業能力的團隊，判斷軟體產品是否符合規劃的需求與各項作業標準，並於審核後提出審查建議(Stark, 2003)，但透過團隊就產品詳細討論後，提供參與者很好的學習機會(Wiegers, 2002)，達到同仁訓練的目的。

(1) 參與人員

有主持人、審查人員、紀錄人員、製作者，團隊審查可由製作者擔任主持人，也可兼報告人身分。

(2) 會議進行方式

團隊審查有五個程序，規劃、預備會議、準備工作、審查會議及修改工作，其與檢視方法中所對應之程序相同，但對產品修改的管制及原因分析，則無結構的程序。

### 2.3.3 逐步審查(Walkthrough)

「逐步審查」是非正式的審查方式之一，由產品作者領導工作團隊或相關團體成員，以集體討論的型態進行 (Doherty, 1997)，由參與者對產品可能的錯誤、不符標準處，提出問題或建議。

#### (1) 參與人員

有製作者、審查人員，並無主持人角色，由被審查產品之製作者選擇同仁參加。

#### (2) 會議進行方式

由產品製作者召集相關同仁，就審查之產品提出說明，由審查人員提出可能的錯誤點。因參加人員少，且不要求事前準備，會議召開時間深具彈性及效率。但根據福特汽車公司的研究資料顯示，運用此方法比檢視方法平均少找出 50%之錯誤點 (Wiegers, 2002)。

## 2.4 同仁審查效益

軟體開發過程中加入同仁審查程序，將增加軟體專案總預算 5% - 15% 的成本 (Gilb, 1993)，其中使用檢視審查方法流程的開發組織，結果絕大多數所獲得效益均高過於成本，最明顯的效益是產品品質獲得提升 (Tervonen, 2005)。分別以錯誤偵測效益、成本效益及其他效益說明如下：

#### (1) 錯誤偵測效益

1976 年 Fagan 指出在系統設計及程式設計階段，使用檢視同仁審查方法，可以偵測出 82% 產品的錯誤。1992 年 Shirey 指出，依照檢視方法實施同仁審查，偵測產品之錯誤比率應在 60-70% 之間。1994 年 Barnard 和 Price 對 AT&T 之研究顯示，檢視審查方法偵測產品之錯誤率可超越 70% 以上 (Harjumaa, 2005)。

#### (2) 成本效益

軟體開發流程中執行同仁審查程序將提升軟體品質，使後續的測試及維護工作減少，開發總成本因而降低。1994 年 Grady 和 Slack 以 Hewlett-Packard 發現使用檢視法執行同仁審查活動，每年節省 33% 開發總成本 (Grady, 1994)。1993 年 Ebenau 和 Strauss 提出案例，接受 NASA 軟體委外處理的軟體公司，評估執行超過 300 次同仁審查活動，因而節省了 7.5 百萬美元的成本 (Harjumaa, 2005)。

#### (3) 其他效益

一般研究者重在同仁審查的經濟效益，包括提升品質、縮短開發時間、降低開發成本等，但同仁審查活動同時也能發揮其他方面的效益，分別說明如下：

- (i) 產品作者透過同仁審查活動獲得工作的回饋，做為改進依據 (O'Hara, 1998)，其他同仁則學習避免錯誤的經驗。
- (ii) 提供同仁在職訓練機會，使其熟悉職務上各項標準、技術、文化及審查的方法 (Doolan, 1992)。
- (iii) 藉由同仁審查活動達到工作互相備援目的 (Doolan, 1992)。
- (iv) 藉由同仁審查活動建立團隊觀念及實際運作 (O'Hara, 1998) (Doolan, 1992)。
- (v) 藉由活動的執行，建立團隊溝通程序與管道。
- (vi) 同仁審查活動的實施，會逐步建立組織在品質方面的文化。
- (vii) 同仁審查活動運用同仁力量協助產品作者提升品質，減少其壓力，因此降低組織離職率。
- (viii) 同仁因審查活動增加經驗及溝通能力，在人力運用上有更大彈性。
- (ix) 同仁審查除

提升軟體品質外，亦可達到凝聚團隊精神、訓練資淺同仁、人員互為備援及建立系統文件的目的，如圖 2 所示。

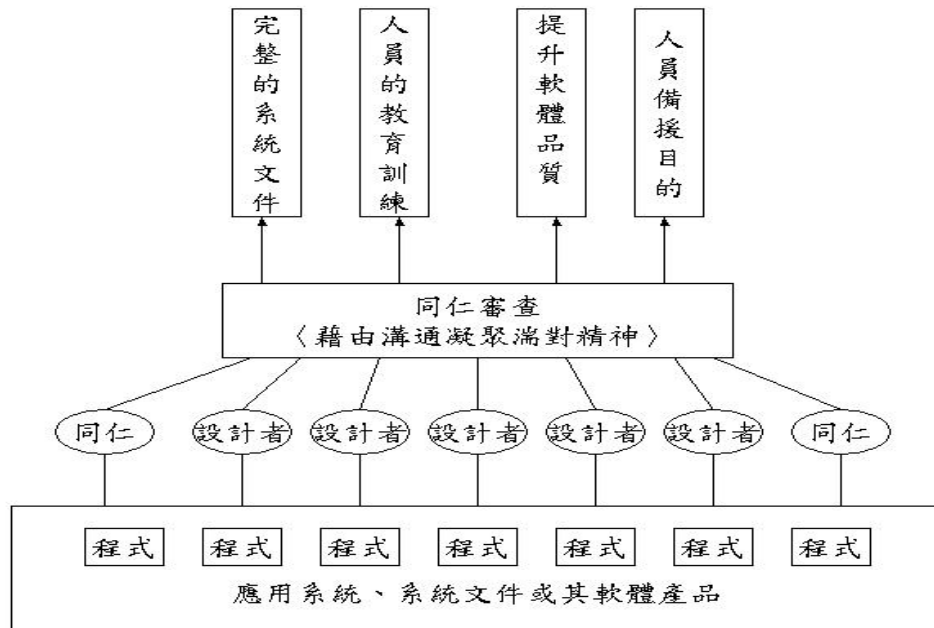


圖2 同仁審查效益圖

## 2.5 執行同仁審查活動成功關鍵因素

依照維基百科全書 (Wikipedia) 解釋，成功關鍵因素 (Critical Success Factors, CSF; Key Success Factors, KSF) 是「一個組織或專案完成任務必要的元素，組織可以使用各種方法識別出邁往成功之路的重要元素」。這些因素影響企業或專案目標的達成，因被執行程度可能造成正面或負面效果。軟體同仁審查活動主要目的是找出軟體產品尚存在錯誤，因此成功的同仁審查活動是能最有效率且發現最多軟體錯誤。同仁審查活動程序與方法主要是以檢視法為基礎，而活動是以會議或討論方式進行，所以檢視法相關程序與會議召開技巧，及管理者的管理都是審查活動重要組成因素。成功關鍵因素的找尋方法據文獻探討各學者專家各有理論，本研究則採用 Saaty (1980) 以分析層級程序法 (Analytic Hierarchy Process, AHP) 分析關鍵成功因素。

### 2.5.1 同仁審查成功關鍵因素

本研究將學者專家所陳述同仁審查成功關鍵因素及應注意事項，及執行同仁審查經驗，彙整並根據其性質將其分為執执行程序、人員能力及技術等三類。

#### (1) 程序類

包括同仁審查之規劃及執执行程序，共分為七項，詳述如下：

##### (i) 詳盡周密的規劃

詳盡規劃是活動成功的一半，尤其審查人員的選定、程序的安排對後續執行績效有很大影響。

##### (ii) 審查人員事前充分的準備

審查人員於審查會議前，詳細閱讀產品相關資料，產品中大部分錯誤即會被發現。

(iii) 設定同仁審查目標

設定同仁審查執行的數據目標，例如：百分之百的需求審查，60%的設計審查，75%的程式審查，50%測試計畫審查或某些重要業務必須實施同仁審查，如證券交易的交易系統各階段都必須實施同仁審查活動等。

(iv) 開發流程中應規劃同仁審查流程

軟體開發專案中常因作業錯誤造成工作必須重複執行，其所耗費之成本占專案總成本的 30-50%(Futrell, 2002)，同仁審查已被證實可早期發現問題，對減少專案中重複工作現象非常有幫助。

(v) 審查活動應盡量正式化

愈結構化的審查活動，獲得效益愈大，因此依照正式的審查辦法實施，會獲得最大效益。

(vi) 設定最佳審查點

同仁審查適用於軟體開發流程各階段之產品及文件之驗證，希望能發現產品錯誤而及早修正，因此在整個開發流程中，產品審查時点的安排會影響審查績效。

(vii) 活動後的績效分析

根據同仁審查執行紀錄，定期分析錯誤發生原因及同仁審查執行效益，以持續改善作業流程、檢查表、紀錄表單及訓練計畫。

(2) 人員能力類

包括相關人員對同仁審查的認知、觀念及能力等項目，共分為七項，詳述如下：

(i) 高階主管的支援

軟體開發專案流程中安排同仁審查活動，必須獲得管理階層支援與督導，其支援方法是在資源、時間、人員及訓練的支援，並定期查閱活動執行績效。

(ii) 適當的審查人員

選擇審查人員必須考慮專業知識與特質，在審查過程中它會影響錯誤發現效益 (Parnas, 1985)。

(iii) 同仁對同仁審查的態度

使團隊同仁熟悉同仁審查目的、功能、運作方式及每一階段開始、完成時機，是審查活動成功條件之一 (Fagan, 2002)。

(iv) 訓練工作

施與團隊成員有關同仁審查方法相關訓練，以加強成員瞭解同仁審查效益及正確執行方法。活動主持人是整個活動關鍵，應加強其會議主持效率、溝通能力及團隊凝聚力 (Vodde, 2006)。

(v) 有經驗的主持人

主持人掌控活動程序、人員安排、運用會議主持技巧，使會議發揮最大效益，因此是活動成功最關鍵人物 (Laitenberger, 1998)。

(vi) 足夠的資源

基於專案時間緊迫、預算有限、人力不足，因而常被犧牲的工作項目就是同仁審查活動，使同仁審查功效無法發揮，導致軟體品質受到嚴重影響。

(vii) 執行力

執行力是企業績效的關鍵之一，也是競爭力的表徵，許多組織的失敗常可歸因於執行力不佳，流於「口號管理」。

(3) 技術類

包括同仁審查各項作業方法及使用工具等項目，共分為七項，詳述如下：

(i) 主管不得參與審查會議

雖然主管工作經驗豐富，但主管不得參加 (Futrell, 2002)，主要是參與審查之同仁會擔心發現產品之錯誤，會令在場主管對產品作者產生不好印象，影響其工作績效考核，所以會故意放慢查錯節奏，導致審查活動效益降低。

(ii) 審查焦點及資源放在重要系統上

O'Hara (1998) 研究建議應專注偵測主要錯誤 (Major)，而非小錯誤 (Minors)，且以投資報酬評斷方式決定同仁審查對象，因此選擇重要的系統實施同仁審查活動效益最高。

(iii) 利用審查工具

為提升審查人員在準備階段，個別閱讀審查文件之效率，檢查表並不能增加審查效益，而是安排不同領域的情境及指引，引導審查者審查方向，可提升作業效能 (Sapsomboon, 1999)。

(iv) 電腦技術的運用

通常同仁審查是以人工方式進行，但在各階段活動中如利用電腦工具處理、傳送、搜尋、備份資料，甚至舉行遠距視訊會議、建立資料庫等業務，都將提升同仁審查效益。

(v) 會議中只專注找出問題，不得討論問題之解決方法

同仁審查活動最為人詬病是耗費成本，因此禁止在審查會議中離題的討論，尤其是花費很多時間討論如何解決問題為最常見 (Johnson, 1993)。

(vi) 全方位的溝通

為因應電腦專業分工及組織架構，同仁審查活動中審查人員挑選必須考慮包含相關專業人員，方能達到審查效果。

(vii) 著重開發作業上游階段

IBM 有關軟體品質的報告指出，在軟體開發初始階段如殘留錯誤，則該錯誤會發生連漪 (Ripple) 作用，在下一階段產生更多錯誤，使品質惡化，並增加作業成本 (Futrell, 2002)，因此著重上游階段的同仁審查，可獲得較高效益。

### 3. 研究模式與流程

本研究藉由前述探討，歸納出同仁審查活動成功關鍵因素，再進行研究如下：

#### 3.1 步驟一：使用德菲法確認關鍵因素

##### 3.1.1 建立關鍵因素構面

經過文獻探討並參酌實務執行經驗，歸納出 21 項同仁審查活動成功關鍵因素，因同仁審查活動包括起始程序、規劃程序、執行程序、控制程序、結束程序，與一般專案



的執行無異，而本研究主要探討同仁審查活動執行效益，因此採用影響專案執行成本、品質、進度最重要的三大元素——程序( Process )、人員( People )、技術( Technical ) (O'Hara, 1998)，做為同仁審查成功關鍵因素分類標準，並依此三個構面將關鍵因素分類成：

- (1) 程序構面：包括規劃的程序、審查人員會議前的準備、設定同仁審查目標、開發流程中應規劃同仁審查流程、正式化的審查活動、設定最佳審查點、活動後檢討分析會議執行效率。
- (2) 人員構面：包括高階主管支持、審查人員的能力、主持人會議主持能力、同仁對品質與同仁審查的態度、主持人及審查人員必須受過同仁審查訓練、足夠的資源（時間與人力）、執行力
- (3) 技術構面：包括系統化的閱讀技術、運用電腦技術、會議中只專注找出問題、主管不得參與、著重開發作業上游階段、審查焦點及資源放在重要系統上、全面的溝通

### 3.1.2 使用德菲法確認關鍵因素

為確認同仁審查活動分類標準及關鍵因素之正確性及合理性，以德菲法利用專家問卷進行確認工作。方法說明如下：

#### (1) 選定受訪專家

德菲法建議由 10-15 位專家組成，每一次問卷調查的對象均固定，不得中途退出，同時各專家不直接進行面對面討論，完全依賴精心設計的系列問卷逐次進行（鄧振源，2002）。本研究選擇熟悉軟體開發生命週期各階段作業，並對軟體品質控制、同仁審查方法實務具有豐富知識或經驗人士參加。

#### (2) 問卷設計

德菲法執执行程序在第一回合調查中，以確認同仁審查分類構面及活動成功關鍵因素為目的，列出程序、人員及技術三分類構面及各關鍵因素。除本研究所列構面及關鍵因素外，專家們也可憑藉專業知識及經驗，提出增訂、修改、刪除意見，及建議理由說明。第二回合以後之問卷設計，視前次問卷回收分析結果設計，並將上次專家反應意見充分表達。

#### (3) 彙集專家意見及回饋控制

德菲法的特色之一是採資訊回饋的調查方式，因此問卷採匿名方式，當問卷回收若專家意見未達一致時，必須彙總專家意見及統計分析之結果，併同新調整問卷調查，再一次發送請專家們確認。

#### (4) 專家意見一致性分析

判斷專家意見趨於一致，可利用平均數、眾數、中位數、四分位差及標準差等方法，由統計資料觀察資料收斂情形。但本研究因調查樣本僅約 10 位，且問卷採用簡單態度尺度(Simple Attitude Scaling)方法，專家只需回答同意或提出其他意見，因此以該問題獲得三分之二專家同意即表示已獲得專家共識。

### 3.2 步驟二：使用分析層級程序法分析資料

為分析關鍵因素及優先順序，採用能以具體的數值顯示各關鍵因素權重的分析層級程序法。其執执行程序如下：

### 3.2.1 建立層級結構

將經專家確認之關鍵因素建立同仁審查活動關鍵因素層級如表 3.1。Saaty(1989)建議使用AHP進行群體決策(Group Decision Making)時，應利用「幾何平均數」做為整合函數，而非以算數平均數。若群體中 $n$ 個決策者，對第一項關鍵因素之權重判斷分別為 $w_1, w_2, \dots, w_n$ ，故以幾何平均數計算「整合函數 $W$ 」，詳如式 3.1。

$$W = \sqrt[n]{w_1 \times w_2 \cdots \times w_n} \quad (3.1)$$

### 3.2.2 彙集專家意見

製作同仁審查關鍵因素調查問卷，藉以向專家調查關鍵因素權值評估資料，問卷是將關鍵因素項目，逐項的以兩兩比較方式列出，依九尺度的名目尺度，由專家們勾選每一要素比較的尺度，問卷必須清楚的敘述每一成對比較的問題。

### 3.2.3 分析關鍵因素權重

本分析分為三個程序，依序說明如下：

#### (1). 建立比較矩陣

決定一群關鍵因素的權重方法是將其兩兩相比，因此將專家對關鍵因素重要性評估資料，依據名目尺度轉換成數據評量資料後，進行每一層活動的成對比較，若某一層中有 $n$ 個關鍵因素，則需進行 $\frac{(n \times n - n)}{2}$ 次的成對比較，並依據成對比較元素值可建立一個 $n \times n$ 階的成對比較矩陣( $A$ )，如式 3.2。

$$A = \begin{bmatrix} \frac{w_1}{w_1} & \frac{w_1}{w_2} & \cdots & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & \frac{w_2}{w_2} & \cdots & \frac{w_2}{w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \cdots & \frac{w_n}{w_n} \end{bmatrix} = \begin{bmatrix} 1 & a_{12} & \cdots & a_{1n} \\ 1/a_{12} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & 1 & \vdots \\ 1/a_{1n} & 1/a_{2n} & \cdots & 1 \end{bmatrix} \quad (3.2)$$

其中 $a_{ij} = \frac{w_i}{w_j} (i, j=1, 2, \dots, n)$ ， $w_1, w_2, \dots, w_n$ 代表 $n$ 個比對的權重值。

#### (2) 計算優勢向量

在獲得成對比較矩陣後，即可利用數值分析中的優勢值(eigen value)解法，求出優勢向量(priority vector)。優勢向量值之計算方法是以將成偶比較矩陣中各列元素相乘，並開 $n$ 次方根( $n$ 為各列元素之總和)，再將開方後之數值標準化(除以開方後之數值)，即得各列所代表目標之優先值。其求取之「優勢向量( $w_i$ )」計算方式，如式3.3。

$$w_i = \frac{\left(\prod_{j=1}^n a_{ij}\right)^{\frac{1}{n}}}{\sum_{j=1}^n \left(\prod_{j=1}^n a_{ij}\right)^{\frac{1}{n}}} \quad (i=1, 2, \dots, n) \quad (3.3)$$

### (3) 一致性檢定

為判斷專家們所作評估的合理程度，必須進行一致性的檢定。Satty (1980) 的建議以一致性指標(Consistence Index, C.I.)與一致性比例(Consistence Ratio, C.R.) 來檢定矩陣的一致性是否必需修正（陳文賢，1991）。一致性檢定執行如下四個步驟：

#### (i) 求解最大特徵值

以成對比較矩陣(A)乘以優勢向量( $w_i$ )，可得新的向量( $w'$ )，詳如式 3.4：

$$w' = A \times w_i = \begin{bmatrix} 1 & a_{12} & \cdots & a_{1n} \\ 1/a_{12} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & 1 & \vdots \\ 1/a_{1n} & 1/a_{2n} & \cdots & 1 \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} w'_1 \\ w'_2 \\ \vdots \\ w'_n \end{bmatrix} \quad (3.4)$$

最大特徵值  $\lambda_{\max}$  計算，詳如式 3.5：

$$\lambda_{\max} = \frac{1}{n} \left( \frac{w'_1}{w_1} + \frac{w'_2}{w_2} + \cdots + \frac{w'_n}{w_n} \right) \quad (3.5)$$

#### (ii) 計算一致性指標

所謂一致性指標 C.I.值之計算式如式3.6

$$C.I. = \frac{\lambda_{\max} - n}{n - 1} \quad (3.6)$$

其中  $\lambda_{\max}$  為最大特徵值，n: 矩陣列數

當 C.I.值 = 0 時，則表示專家們對各項因素前後判斷句一致性，而其值越大表示不一致性越高，而 Saaty (1980) 建議 C.I.值  $\leq 0.1$  為可容忍的誤差範圍。

#### (iii) 計算一致性比率

矩陣數目及名目尺度皆會影響 C.I.值的大小，在正倒值矩陣數及名目尺度已知下，隨機產生 C.I.的稱為隨機指數(random index, R.I.)。R.I.與矩陣數有關，如表 3.2，而一致性比率 C.R. 之計算公式如下：

各層級一致性比率 C.R.計算如下式，係一致性指標 C.I.除以隨機指標而得。

$$C.R. = \frac{C.I.}{R.I.} \quad (3.7)$$

若C.R. > 0.1，則檢定值視為不可接受。若C.R.  $\leq 0.1$ ，則檢定值視為可接受。R.I. 會隨矩陣階數增加而增加。

### 3.3 步驟三：建議個案同仁審查改善方案

非正式同仁審查沒有標準流程與規定，是由使用者視需要來決定做法，也因此容易忽略了重要關鍵因素之程序或項目，本研究根據分析成功關鍵因素及其重要性，以個案研究案例——證交所系統設計所實施之非正式審查流程為基準，調整其同仁審查流程及執行應注意事項，供證交所及一般中小型軟體開發組織參考。

### 3.4 步驟四：驗證改善效果

為評估建議流程之合宜性及具預期之效益，以專家訪談方式進行驗證工作，訪談的對象是個案負責軟體品質管制之制度訂定人員，及一位熟悉同仁審查方法之專家。訪談方法是先說明整個研究目的及方法，並將同仁審查關鍵因素重要性分析結果、個案原執行之非正式審查流程及建議修正後之審查流程等書面資料詳加說明，最後請其就新流程之可行性及對品質、成本、時間改善程度等資料加以評估，並請其提出建議，如驗證結果符合研究目的，則可提出本研究之建議與結論，如驗證結果與研究目的不符，必須探究原因，如原因係研究基礎錯誤，則必須回到步驟一，重新開始研究工作。

## 4. 個案研究

本研究將前述建構模式與流程應用到證交所之系統設計階段中同仁審查活動。

### 4.1 個案背景說明

#### (1) 軟體審查流程

台灣證交所交易系統之開發專案視系統重要性，分別在需求確認、系統設計、程式設計及系統文件（包括重要測試計畫）階段實施同仁審查活動，系統設計是開發流程中最重要工作，其產品與程式設計、測試、操作及系統資源規劃等作業息息相關，因此審查活動更顯重要，除先執行設計團隊審查外，另需安排跨單位審查活動，本研究也以其為對象，探討其流程與詳細工作說明如下，執行流程圖如圖 3：

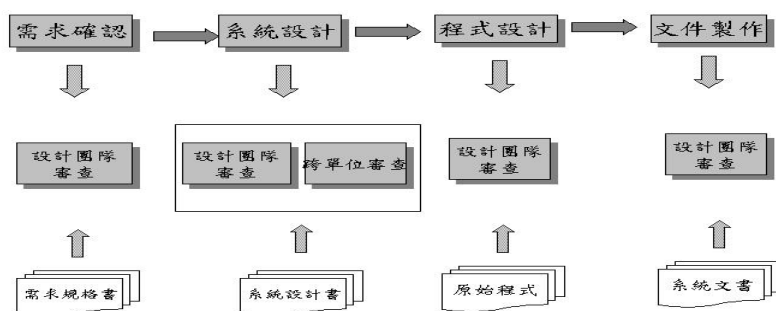


圖 3 證券交易所同仁審查流程圖

#### (i) 設計團隊審查

交易系統開發過程中，系統設計完成後，系統設計團隊負責人召集系統設計人員、共同維護人員進行產品審查討論。活動由團隊負責人擔任主持人，系統設計人員報告需求、系統架構、檔案或資料庫架構、程式說明、執行方式、安全控制機制、備援功能等項目；審查人員就不正確或不完善部分提出問題。活動未做任何會議紀錄。討論中未使用檢查表或其他參考的閱讀技術文件。會後設計人員依決議事項進行產品修正工作。設

計團隊並無任何審查技術訓練，新進人員也未實施審查程序說明。

## **(ii) 跨單位同仁審查程序**

設計團隊審查後，另召開跨單位的同仁審查會議，由系統需求、系統開發、系統維護及使用各單位共同審查。跨單位同仁審查較正式，會議前相關資料先送至參加會議單位，由主管指派承辦人員參加。會議的進行由規劃部主管或副主管主持，設計團隊負責報告，會議中發現錯誤也會一併討論解決方法。每次會議參加人數約 15-20 人。召集會議的團隊會留下正式會議記錄。

## **4.2 問題分析**

證交所使用的同仁審查方法，經過多年運作，已能達到品質驗證效果，但為提升其執行績效，期望藉由目前運作程序之探討，繼續改進。依據比較結果及曾實際參與作業了解之情形，本研究整理出目前交易所同仁審查活動可能影響執行績效的問題點：

### **(1) 審查活動應盡量正式化**

證交所目前規定線上交易系統或系統負責人認為重要系統，必須執行同仁審查程序，但所制訂之「設計審查作業指導書」，僅規範書面審查規定，有關同仁審查時機、程序及方法並未定義，目前是依照同仁經驗執行。

### **(2) 未要求審查人員會議前充分的準備**

證交所審查程序中，未要求審查人員於會議前，必須對產品相關文件詳細準備。

### **(3) 審查時機**

證交所程式設計完成後之同仁審查時機，安排在系統測試完成後方執行。

### **(4) 未實施同仁審查訓練工作**

證交所未對審查會議的主持人給予適當的訓練，對系統設計及程式設計人員也未安排審查方法的說明，只是依照一般會議方式進行，無法充分發揮同仁審查效益。

### **(5) 未建立審查檢查表**

交易所從事資訊工作同仁，因長年服務同一崗位，對負責系統已瞭若指掌，因此未建立審查表，使審查經驗無法累積，容易造成審查人員疏漏。

### **(6) 未評估執行績效**

任何活動執行完畢均應檢討，做為下次活動改進參考，交易所同仁審查程序中，未完全紀錄審查活動結果，亦未制定活動完成後之績效評估程序。

## **5. 研究結果與案例驗證**

本研究根據研究方法，逐步完成研究工作，並另尋專家協助結果驗證。

### **5.1 德菲法調查結果**

第一次問卷及第二次問卷結果，專家對同仁審查各項成功關鍵因素均達到共識。修正後的關鍵因素如下：

- (1) 程序構面：包括規劃的程序、審查人員會議前的準備、設定同仁審查目標、開發流程中應規劃同仁審查流程、正式化的審查活動、活動後檢討分析會議執行效率。
- (2) 人員構面：包括高階主管支持、審查人員的能力、主持人會議主持能力、同仁對品質與同仁審查的態度、主持人及審查人員必須受過同仁審查訓練、足夠的資源

(時間與人力)、執行力。

- (3) 技術構面：包括完善的檢查表、運用電腦技術、會議中除簡單問題外不可討論解決方法、著重開發作業上游階段、審查焦點及資源放在重要系統上、全面的溝通。

## 5.2 以層級分析程序法分析結果

依據「同仁審查成功關鍵因素層級架構」，設計層級分析程序法專家問卷，並據以分析，獲得各關鍵因素之整體權重值（如表 1）。藉由整體權重值之排列，了解同仁審查活動執行應注意事項及做為建議個案同仁審查改善方案之重點參考。

表 1 同仁審查關鍵因素重要性排序表

關鍵因素	整體權重	排序
完善的檢查表	<b>0.087</b>	<b>1</b>
規劃的程序	<b>0.082</b>	<b>2</b>
開發流程中應規劃同仁審查流程	<b>0.074</b>	<b>3</b>
審查人員會議前充分的準備	<b>0.068</b>	<b>4</b>
設定同仁審查目標	<b>0.065</b>	<b>5</b>
審查焦點及資源放在重要系統上	<b>0.063</b>	<b>6</b>
有經驗的主持人	<b>0.056</b>	<b>7</b>
訓練工作	<b>0.055</b>	<b>8</b>
全面的溝通	<b>0.050</b>	<b>9</b>
適當的審查人員	<b>0.050</b>	<b>9</b>
著重開發作業上游階段	<b>0.048</b>	<b>10</b>
執行力	<b>0.048</b>	<b>10</b>
足夠的資源	<b>0.047</b>	<b>11</b>
同仁對品質與同仁審查的態度	<b>0.046</b>	<b>12</b>
會議中除簡單問題外不可討論解決方法	<b>0.038</b>	<b>13</b>
正式化的審查活動	<b>0.036</b>	<b>14</b>
高階主管的支持	<b>0.035</b>	<b>15</b>
活動後檢討分析會議執行效率	<b>0.033</b>	<b>16</b>
運用電腦技術	<b>0.027</b>	<b>17</b>

## 5.3 建議改善方案

將研究結果與交易所流程比較，各關鍵因素實施狀況如表 2，據此資料及針對同仁審查活動之問題，提出建議改進方法如下表：

表 2 個案關鍵因素實施狀況比較表

成功關鍵因素	排序	個案實施狀況	說明
完善的檢查表	<b>1</b>	<b>X</b>	

規劃的程序	2	V	設計團隊審查未執行
開發流程中應規劃同仁審查流程	3	V	
審查人員會議前充分的準備	4	X	未明文規定
設定同仁審查目標	5	V	
審查焦點及資源放在重要系統上	6	V	交易系統必須實施
有經驗的主持人	7	V	
訓練工作	8	X	
適當的審查人員	9	V	由主管指派承辦人員參加
全面的溝通	9	V	透過跨單位審查活動
執行力	10	V	
著重開發作業上游階段	10	V	注重需求確認及系統設計結果審查
足夠的資源	11	V	
同仁對品質與同仁審查的態度	12	V	
會議中除簡單問題外不可討論解決方法	13	V	
正式化的審查活動	14	X	
高階主管的支持	15	V	
活動後檢討分析會議執行效率	16	X	
運用電腦技術	17	V	

V：表示已實施

X：表示尚未實施

#### (1) 簡化審查方法

「規劃程序」項目改善建議，證交所交易系統或重要應用系統設計完成後，均執行兩次審查活動—著重技術的設計團隊審查及著重溝通的跨團隊審查，造成設計團隊時間及人力的浪費，建議簡化審查活動成一種方式，並制定審查辦法，可增加設計團隊工作效率、縮短作業時間，且因有明確的審查辦法而易於規劃、執行與管理。

#### (2) 改善審查流程

交易所設計團隊審查未執行規劃的程序、準備會議、活動後績效評估等項目，跨單位審查未執行準備會議、活動後績效評估等項目。由於交易所同仁經驗豐富的特質及溝通方便，準備會議確無召開必要，雖然本研究研究結果中，「審查後績效評估程序」並非極重要關鍵因素，但依據 CMMI 模式，要成為「能力成熟」的組織，數據化的管理是必要的，因此建議新的審查流程審查程序包括規劃的程序、會議前準備工作、審查會議、修改工作、決議追蹤及會議績效評估等項目(審查流程圖如圖 4)。另根據研究結果，審查會議的召開並非重要關鍵因素，因此對於小型開發案與維護案，則可將建議流程中之審查會議省略，由設計團隊負責將審查人員審查結果彙整，並據以修改產品，其他流程不變，各程序分別說明如下。

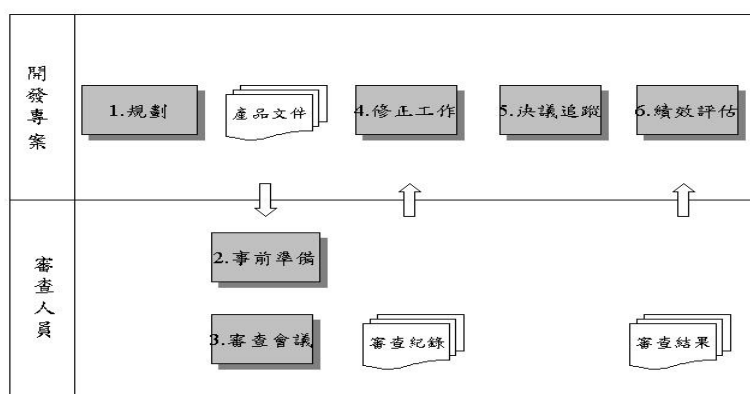


圖 4 建議交易所實施新同仁審查流程圖

#### (i) 規劃的程序

根據同仁審查關鍵因素「著重開發作業上游階段」、「審查焦點及資源放在重要系統上」之原則，建議訂定同仁審查程序書，包括審查流程各工作項目、起始條件、結束條件及文件格式等，供未來執行軟體專案時，遵循規劃同仁審查工作。

#### (ii) 會議前的準備

建議未來審查活動必須明確指派適當審查人員，且強制要求審查人員審查前必須詳實閱讀受審產品文件。為協助審查人員發揮最大效益，完善的檢查表是必須的。訂定情境基礎 (Scenario-Base) 閱讀技術，指導審查人員如何發現資訊，如何從資訊中發現問題 (Biffel, 2003)，提醒審查者注意事項，比提供檢查表更有效率。

#### (iii) 審查會議

建議審查會議人數縮減為約 10 人內，以提升效率，有關經驗傳承及訓練工作應安排以其他方式完成，明確界定審查會議主要目的是找出產品錯誤。

#### (iv) 修改工作與決議追蹤（修改工作、決議追蹤兩項流程與現行相同）。

#### (v) 活動後的績效評估

為持續改善同仁審查活動，每次活動主持人、審查人員及所耗費時間、人力、發現錯誤數都應詳細紀錄、分析，做為活動改善或訓練安排之參考。

### (3) 審查權責確立

目前證交所未建立獨立審查人員制度，容易造成權責不清狀況，建議審查人員及審查會議獨立於開發專案之外，明確賦予審查人員責任。審查權責由專責人員負責，審查人員之遴選不再由參與會議單位指派，將符合關鍵因素「適當審查人員」之項目。

### (4) 訓練工作

同仁審查關鍵因素中「適當的審查人員」、「有經驗的主持人」、「同仁對品質與同仁審查的態度」、「主持人及審查人員必須受過同仁審查訓練」等四項，都可藉由訓練工作完成，為提升同仁審查效益，必須對同仁進行專業訓練，包括主持人、審查人員及一般資訊人員。新進員工訓練時，也要使其明瞭同仁審查執行時機及程序。負責軟體品質管理同仁，對國內外有關軟體品質新知應掌握，以便運用達到持續改善的目的。



#### (5) 建立基本制度

CMMI 模式中定義 Level 3 等級組織必須具備驗證流程，而同仁審查是驗證流程中必須完成的特定目標之一。依 CMMI 模式之定義，組織進行 Level 3 流程之前，必須先建立完成 Level 2 流程制度，包括需求管理流程、專案管理流程、度量與分析流程、流程與產品品質保證流程、型態管理流程等，因此交易所應評量基本制度是否已配合建立完善，方不致影響同仁審查的執行效益。

#### 5.4 結果驗證

本研究另訪談兩位專家以確有研究結果之可行及待加強之處。兩位均認為本研究觀察個案同仁審查活動缺點，非常客觀與確實，所提改善措施亦屬可行，他們相信實施改善措施必定會提升同仁審查效益；但一位專家認為建議第五項（CMMI Level 2 流程制度）無法掌控外，其他建議可逐步實施。

表 3 同仁審查改善意見驗證訪談表

意見類別	建議事項	專家意見
程序	<ol style="list-style-type: none"> <li>整合審查方法 將設計團隊審查與跨單位審查二種方式整合成一種審查方式，以增效率。</li> <li>改善審查流程 確定規劃、事前準備、審查會議、修正產品、工作追蹤及績效分析六程序。小型開發或維護案可省略審查會議之召開。</li> <li>會議前審查人員之準備 應強制要求審查人員會議前必須確實閱讀完畢審查文件，並紀錄準備時間。</li> <li>訂定同仁審查程序書 供未來執行軟體專案時，規劃同仁審查工作遵循標準。</li> <li>增加績效分析工作 發現錯誤數都應詳細紀錄、分析，做為活動改善或訓練安排之參考。</li> <li>權責區分 將審查工作由專任品管人員負責，由其安排主持人及審查人員。</li> </ol>	<ol style="list-style-type: none"> <li>兩位專家均同意整合後可增加效率。</li> <li>兩位專家均同意建議之審查流程。</li> <li>一專家認為其同仁會自行準備，但也同意強制要求之規定。另一專家認為除強制要求外，還必須規定考核辦法。</li> <li>兩位專家均同意訂定同仁審查程序書</li> <li>兩位專家均同意</li> <li>兩位專家均同意</li> </ol>
人員	<ol style="list-style-type: none"> <li>訓練工作 為提升同仁審查效益，必須對同仁進</li> </ol>	<ol style="list-style-type: none"> <li>一專家：將安排內部訓練及新進員工訓練課</li> </ol>

	<p>行同仁審查專業訓練，包括主持人、審查人員及一般資訊人員。</p> <p>2. 減少審查參加人員</p> <p>3. 適當的審查人員 應避免單位指派，由專任品管人員負責安排適當審查人員。</p>	<p>程。另一專家：訓練工作必須安排。</p> <p>2. 兩位專家均同意應減少審查參加人員在10人以下。</p> <p>3. 兩位專家均同意應避免由單位指派審查人員。</p>
技術	<p>1. 檢查表 建立情境基礎閱讀技術，指導審查人員如何發現資訊，如何從資訊中發現問題，提醒審查者閱讀審查文件需注意事項等。</p>	<p>1. 一專家：可行的是先建立完善的檢查表，有關建立情境基礎閱讀技術，將蒐集資料再評估如何建立。另一專家：可先從訓練著手，再選擇較小的專案試行。</p>

## 6. 結論及建議

### 6.1 研究結論

本研究將 19 項同仁審查活動成功關鍵因素分為程序、人員與技術等三構面，經過嚴密的研究程序，獲得研究結果說明如下：

#### (1) 同仁審查組成構面

本研究經由整合及歸納專家共識意見後，以「程序」、「人員」及「技術」等三項構面，進行同仁審查活動關鍵因素的評量。結果資料顯示三構面幾乎同等重要，只有些微差異。其中「程序」為同仁審查中較重要構面，這也符合同仁審查強調愈正式的程序愈有效益。「人員」的構面又比技術重要，因此組織內同仁必須要培養品質與審查觀念，並注重同仁審查訓練工作，這些也是中小型開發組織所忽略的。

#### (2) 重要同仁審查關鍵因素：

在 19 項同仁審查成功關鍵因素中，前十名重要因素依序為「完善的檢查表」、「規劃的程序」、「開發流程中應規劃同仁審查流程」、「審查人員會議前充分的準備」、「設定同仁審查目標」、「審查焦點及資源放在重要系統上」、「有經驗的主持人」、「訓練工作」、「全面的溝通」及「適當的審查人員」等，可作為執行同仁審查方法之參考。

#### (3) 同仁審查活動修正方案

軟體開發組織基於資源或產品品質需求狀況，安排不同方式的同仁審查流程；非正式的流程可根據本研究結果制定或參考重點修正執行，其重點在健全同仁審查流程、審查業務權責應獨立、建立完善檢查表、加強同仁審查工作之訓練及執行注意事項，這些改善措施亦可供其他軟體開發組織借鏡。

### 6.2 研究貢獻

歸納本研究之貢獻有以下三點，說明如下：

(1) 本研究歸納同仁審查關鍵因素，又經管理者及軟體品質實際作業人員以德菲法確認，這些因素實為同仁審查核心，可供軟體開發組織制定或檢討審查制度之參考。

(2) 本研究利用分析層級程序法，藉由專家評估完成以具體量化數據值呈現所有關鍵因素優先順序。可供中小型開發組織規劃依本身資源及需要，選擇重要關鍵因素實施非正式同仁審查流程及執行之參考，使其在有限資源下，實施同仁審查活動以獲最大效益。

(3) 國內軟體開發組織除了通過 CMMI Level 3 以上等級認證的 13 家軟體廠商及組織外，對同仁審查活動大都未制定執执行程序或辦法，本研究結果可使中小型軟體開發組織了解同仁審查目的、執执行程序及重要關鍵因素，進而於開發流程中規劃審查活動以提升軟體品質。

### 6.3 未來研究方向

本研究雖順利完成關鍵因素及重要性分析，並針對個案建議同仁審查流程，但宥於時間、人力等問題，仍有部份未盡週詳之處，因此提出下列二項建議，供有興趣對軟體同仁審查繼續研究之參考。

(1) 「軟體同仁審查」是以「人工」審查方式為主，近年來以軟體工具或人工智慧方法取代人工審查之研究正方興未艾，或提供未來研究者研究方向。

(2) 國外專家學者對軟體同仁審查領域已有豐富的研究結果，但國內研究卻非常少。因教育背景與組織文化的差異，其研究結果是否完全適用於國內組織？是否需要調整？亦可做為未來研究方向。

## 參考文獻

### 一、中文部份：

經濟部 (2004)，我國CMM/CMMI應用現況與發展趨勢，資策會，2004，台北

資策會 (2002)，國際分工及價值創新是台灣資通訊產業全球競爭必走之路

劉小鳳 (2003)，淺談軟體品質，品質月刊，2003 年 11 月，39:11, pp.95-99

張文貴(2000)，全面軟體品質管理的實施策略（下），品質月刊，2000 年 9 月，36:09, pp.43-48

鍾樹人(2006)，軟體別再出錯了，科學人雜誌，2006 年 7 月，P.85-96

飛思科技 (2005)，軟體測試理論與實作，博碩文化，台北。

趙志宏 (2006)，武器獲得專案管理能力成熟度評量模式之研究，華梵大學工業工程與經營資訊學系碩士論文，台北。

吳碧玉 (2003)，民宿經營成功關鍵因素之研究-以核心資源觀點理論，碩士論文，朝陽科技大學企業管理研究所。

鄧振源、曾國雄 (1989)，分析層級法(AHP)的內涵特性與應用(上),中國統計學報,27(6), P.5-22。

林建煌編譯 (2002)，現代管理學，華泰，台北。(S. P. Robbins (2001), Fundamentals of Management)

林信惠、黃明祥、王文良 (2005)，軟體專案管理，智勝出版社，台北。

陳文賢 (1991)，管理科學—作業研究與數學方法，三民書局，台北。

遠擎顧問公司譯 (2002)，策略核心組織，臉譜出版，台北。(R.S.Kaplan (2001), The Strategy-Focused Organization)

### 二、英文部分：

Bailey, P.M. (2004), A Formula for Successful Peer Reviews, Better Software Magazine, December 2004, p. 36

Biffi, S. (2003), Investigating the Defect Detection Effectiveness and Cost Benefit of Nominal Inspection Teams, IEEE Transacation on software engineering, vol 29, no.5, May 2003

- Boehm, B. (1981), Software engineering economics, Prentice-Hall advances in computing science and technology series, Englewood Cliffs, N.J.
- Boehm, B., Basill V. (2001), Software Defect Reduction Top 10 List, IEEE Computer, vol 34, no 1, pp.135-137.
- Chao, L.P., Ishil K. (2004), Design Process Error-proofing Engineering Peer Review Lesson From NASA, ASME Design Engineering Technical Conferences, October 2004, Salt Lake City, Utah.
- Dhaliwal, I.S, Tung I.I. (2000), Using Group Support Systems for Developing a Knowledge - Based Explanation Facility, International Journal of Information Management, P. 131-149.
- Doolan, E.P. (1992), Experience with Fagan's Inspection Method, Software – Practice and Experience, vol.22, February 1992, pp.173-182
- Futrell, R.T. (2002) , Quality Software Project Management, Prentice Hall PTR, NJ, pp.878-885 .
- Fagan M. (1976), Design and Code Inspections to Reduce Errors in Program Development. IBM Systems Journal 3, pp182-210.
- Fagan, M. (2002), A History of Software Inspection, software pioneers Eds: M.Broy, E.Denert, Springer 2002, pp.215-225.
- Freimut, B., Klein B., Laitenberger O., Ruhe G. (2000), Measurable Software Quality Improvement through Innovative Software Inspection Technologies at Allianz Life Assurance. <http://www.info.uni-karlsruhe.de/lehre/2004SS/swk/Papiere/escom-paper.pdf>, p. 4
- Gilb, T, Graham D. (1993), Software Inspection, Addison-Wesley, 1993, p.4.
- Grady, B., and Slack, V. (1994), Key Lessons in Achieving Widespread Inspection Use, IEEE Software, Vol. 11, No. 4.
- Harjumaa L. (2005), Improving the Software Inspection Process with Patterns, Faculty of Science, Department of Information Processing Science, University of Oulu, OULU 2005.
- Herbsleb, J., Zubrow D., Goldenson D., Hayes W., Paulk M.(1997), Software Quality and the Capability Maturity Model, Communications of the ACM, Vol 40 , Issue 6, June 1997, pp.30-40
- Laitenberger, O. (1998), An Encompassing Life-Cycle Centric Survey of Software Inspection, Journal of system and software, vol 50, no 1, January 2000, pp5-31
- O' Hara, F. (1998), Optimising Peer reviews/Inspections for individuall learning and team building , [http://www.insight.ie/downloads/optimising\\_inspections.doc](http://www.insight.ie/downloads/optimising_inspections.doc).
- O' Neill, D. (2001), Peer Reviews. Encyclopedia of Software Engineering, New York, New York: Wiley Publishing, Inc..
- Pannnas, D.L., Weiss D.M. (1985), Active Design Reviews, Principles and Practies, International Conference on Software Engineering, p. 132
- Ranganathan, K. (2001), How To Make Software Peer Reviews Work, Quility Progress, Feb 2001,pp.69-72
- Saaty, T.L. (1980), The analytic hierarchy process, McGraw-Hill, New York.
- Saaty, T.L., (1998), The Analytic Network Process: Decision Making With Dependence and Feedback. RWS Publications, Pittsburgh, PA.
- Saaty, T.L. (2001), Decision Making for Leaders: the Analytical Hierarchy Process for Decisions in a Complex World, RWS Publications, Pittsburgh, PA.
- Sapsomboon, B. (1999), Software Inspection and Computer Support, <http://scholar.google.com/scholar?hl=zh-TW&lr=&q=Sapsomboon%2C+1999&btnG=%E6%90%9C%E5%B0%8B&lr=>.
- Schaefer, H. (2000), Fast Software Development Needs Fast Review Techniques, P.1
- Stark, J.(2001), Peer Reviews in Open-Source Development, [http://ecommerce.cit.gu.edu.au/cit/docs/theses/JStark\\_Dissertation\\_OSS.pdf](http://ecommerce.cit.gu.edu.au/cit/docs/theses/JStark_Dissertation_OSS.pdf).
- Tervonen, I. (2002), Software inspection – a blend of discipline and flexibility, <http://www.tol.oulu.fi/~tervo/Rome.pdf>.
- Tjahjono, D. (1996), Exploring The Effectiveness of Formal Technical Review Factors With CSRS, a collaborative software review system. PhD thesis, Department of Information and Computer Science, University of Hawaii.
- Vodde, B., LU Y. (2006), An Inspection Improvement Story: Hangzhou R&D, software quality managment, SQP. Vol. 8, No. 2, 2006, pp. 5-10
- Wieggers, K.E. (2002), Peer Reviews in Software — A Practical Guide , Addison-Wesley, Boston.