

教學個案

◎軟體保證系統：架構與個案之探討

◎標準作業程序在醫療產業的應用—以中央健保局高屏分局為例

軟體品質保證系統： 架構與個案之探討

A Study of Software Quality Assurance System:
Architecture and Case Study

呂執中 *Jr.Jung Lyu*

國立成功大學

National Cheng Kung University

周福星 *Fu-Shine Peter Chou*

國立成功大學

National Cheng Kung University

吳錫賢 *Hsi-Hsien Wu*

國立成功大學

National Cheng Kung University

摘 要

近年來軟體品質越來越受到重視。軟體業者為提昇品質，正積極推動如 ISO 9000 之品保系統。在推動 ISO 品保標準驗證過程中，ISO 9001 之 4.20 與 ISO 9000-3 之 6.4 要求業者須運用統計技術及進行流程與產品衡量。不過目前在軟體衡量之論文中，並未同時考量軟體開發組織之要求與流程的變異程度，故無法直接顯示出流程達成目標的程度。本研究結合軟體流程改善與能力評定 (SPICE) 國際品保系統標準與製程能力分析，針對軟體開發的特性發展一軟體品保系統之架構。此外，經由實際的個案探討可發現，軟體開發過程中，針對關鍵軟體流程的特性，軟體開發組織有必要運用統計製程管制方法，加以隨時監控，以提昇其品質保證之能力。

關鍵詞：軟體品保系統、統計製程管制、製程能力

Abstract

The importance of the software quality is well recognized. Recently, many software developers have established the ISO 9000 quality assurance systems in order to improve the software quality. To satisfy one of the requirements in the ISO 9001 standard (4.20) or ISO 9000-3 (6.4), a software developer must use statistical techniques to measure the quality of its processes and products. However, the surveyed literature regarding software metrics has not studied the way to coordinate the requirements of the software developers with the variation in the process and failed to use a clear index to identify the degree of how a process meets the target requirements. This research integrates the SPICE (Software Process Improvement and Capability dEtermination) international standard and capability analysis to develop an architecture for a software developer to implement the quality assurance system. Based on the case study, the feasibility of the proposed architecture is justified. This study concludes that a software developer requires to apply statistical process control to its key processes in order to monitor and improve its software quality.

Key words: Software Quality Assurance System, Statistical Process Control, Capability

壹、緒 論

在日趨競爭的經濟環境中，由於資訊科技的快速進步，企業對於資訊科技的依賴日益增加。相對的，亦有愈來愈多的軟體問世或資訊系統的開發，來滿足企業的需求。品質不佳的軟體，可能造成企業無法正常運作，導致顧客抱怨、財務損失或商機錯失。因此，「軟體品質」已成為軟體需求者重視的項目，如何開發出品質佳的軟體，便成為值得探討的課題。

爲了能開發出品質優良的軟體或資訊系統，國內目前有資策會與鼎新電腦等機構與企業，積極地推動軟體品保系統並已獲得 ISO 9000 品保驗證。在推動 ISO 9000 軟體品保系統過程中，ISO 9001 之 4.20 統計技術，以及 ISO 9000-3 「ISO 9001 應用於軟體之開發、供應與維護之指導綱要」的支援性活動之 6.4 評量，要求必須進行產品與流程的評量，但是該如何進行評量則沒有具體的說明。不過目前軟體業流程的衡量之文獻中 (如 Clapp, 1993; Costello & Liu, 1995)，多未考量顧客或組織預定之要求的概念，以及流程

的變異程度。因此，其衡量指標並無法直接顯示出軟體開發流程，達成顧客要求或組織預定目標的程度，以及流程的變異程度。同時，在目前國內的軟體業界，對於在評定一個軟體開發公司是否有能力來承接軟體開發專案，並沒有一個具體完整的方法來評定。

針對上述情形，本研究將結合軟體流程改善與能力評定 (SPICE) 國際品保標準與統計制程管制之能力的概念於軟體開發的過程，以達到：(1) 結合 SPICE 標準以建立軟體流程衡量架構，在軟體開發過程中，建立製程能力指標的衡量，以作為軟體開發組織監控軟體流程水準、持續改善與品質稽核的基礎。(2) 以實際案例，來說明本研究發展之軟體製程能力指標衡量架構的運用。

本文架構如下：第貳節將介紹品保系統之標準，第參節提出軟體品保系統之架構，第肆節以一個案說明此架構及流程能力如何衡量，第伍節則做一綜合整理，並對未來發展方向提出建議。

貳、軟體品保系統

軟體品保模式是協助軟體業建立品質管理系統的標準。目前所發展之軟體品保標準，除了 ISO 9000 品保標準外，還有英國所發展的 Tick IT 體系、美國卡內基-美隆大學軟體工程學院發展之軟體成熟度模式 (CMM)，以及由 ISO / IEC 發展的軟體流程改善與能力評定 (SPICE) 標準。以下將針對這四個軟體品保系統，作一介紹。

一、ISO 9000 國際品保標準

ISO 9000 國際品保標準首先是由歐體率先採用，以建立一套歐洲共同市場的一致品質標準，以便讓歐體各國間或世界各國與歐體進行貿易或合作生產時，在品質的要求上能夠有所依循。1987年版的 ISO 9001 - ISO 9003 標準主要是考慮製造業的情況，不過當這標準愈往非製造業推動，愈發現其不足以應付各種產業的情況。ISO / TC 176 技術委員會在 1991 年針對軟體產業，編訂了 ISO 9000-3 「ISO 9001 應用於軟體之開發、供應與維護之指導綱要」，提供軟體業推動 ISO 9001 時的參考。

ISO 9000-3 指導綱要的主體為第四、五、六等三大部份，如圖 1 所示。其

中第四部份是以整體的觀點探討 ISO 品保系統之特點，除了明定管理階層之責任與文件化之品質系統外，更藉著內部品質稽核與矯正措施來保持系統的活力。第五部份是以資訊界慣用的軟體系統生命週期來討論各階段要注意的項目。而第六部份則是與階段性工作無關的支援性活動 (呂執中等, 1994)。

二、Tick IT 體系

Tick IT 體系是英國在 1991 年成立的，主要的目標在於建立一個授權體系。凡經授權的認證機構，必須遵守一致性規定並派用有經過訓練且資格認證的評審員，來進行軟體的評審工作。Tick IT 體系可提供軟體品質系統認證服務而且是符合軟體工業的需求 (Johnson 等, 1995)。Tick IT 認證所依據的是 ISO 9001 / 2 的要求，指導性文件為 Tick IT Guide，它同時也涵蓋 ISO 9000-3 的規範要求，如圖 2 所示。

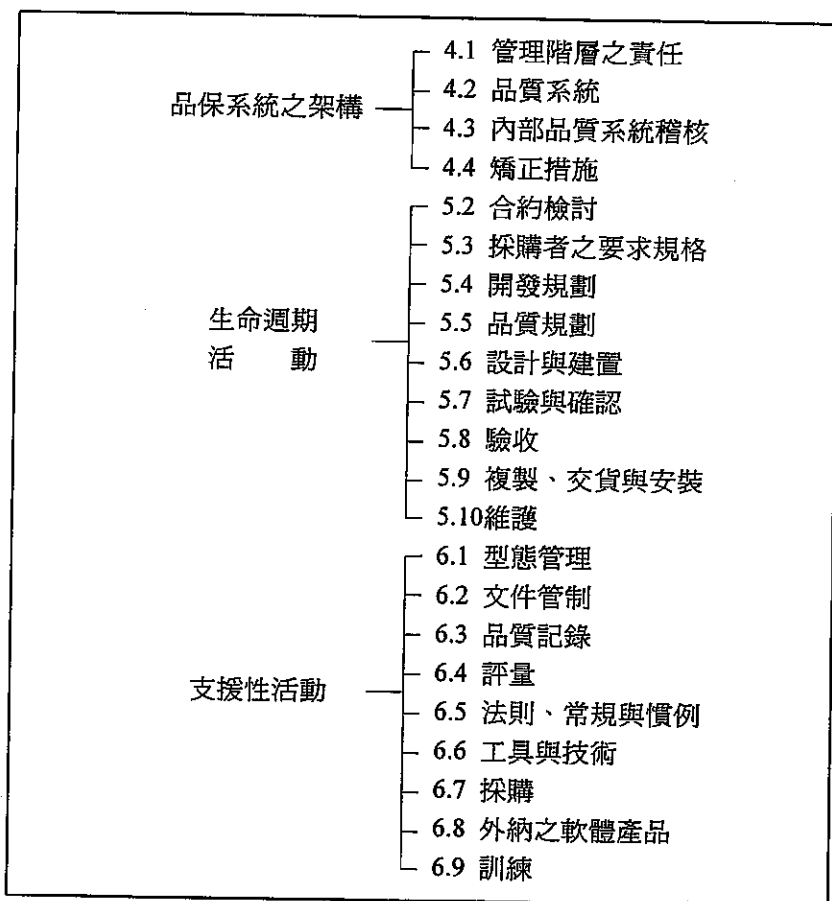


圖 1 ISO 9000-3 指導綱要

三、能力成熟度模式 (Capability Maturity Model, CMM)

能力成熟度模式 (CMM) 是由美國卡內基－美隆大學 (Carnegie-Mellon University, CMU) 的軟體工程學院 (Software Engineering Institute, SEI)，針對軟體生產流程發展出來，以作為全面品質管理與流程改善的架構。這個模式使用起始層 (Initial)、規律層 (Repeatable)、定型層 (Defined)、監管層 (Managed) 與最佳層 (Optimizing) 等五個流程成熟度階層，來評估組織的成熟程度，並針對不同的成熟程度，提供流程改善的策略，其架構如圖 3 所示 (Freese 等, 1994)。

<u>Tick IT 供應商品質系統</u>	
<u>品質系統要項</u>	<u>品質管制要項</u>
4.1 實施與維持品質系統	5.1 鑑定任務
4.2 管理組織與介面	5.2 製作建議書
4.3 審查 (設計)	5.3 提舉專案
4.4 文件證明專案與產品	5.4 分析要求與規定系統
4.5 遴用、訓練與人員發展	5.5 生產高階軟體設計
4.6 建構管理	5.6 生產細部軟體設計
4.7 支援、安全與建檔	5.7 程式製作與單元測試
4.8 品質系統審查與稽核	5.8 整合與系統測試
4.9 採購	5.9 實施允收測試
4.10 進程監控與報告	5.10 移交／釋出
4.11 不合格物料	5.11 使用者訓練
4.12 矯正措施	5.12 儲存與裝運
4.13 品質資訊	5.13 複製
	5.14 交貨與安裝
	5.15 維護與加強
	5.16 支援
	5.17 逐步淘汰

圖 2 Tick IT 指導綱要

	階段	特徵	關鍵流程範圍	品質方法	結果
流程成熟度	5 最佳層 (Optimizing)	改善回饋到流程中	改變技術, 問題分析	從流程變異最小化	品質與生產力 (低) 風險 (高)
	4 監管層 (Managed)	評量流程 (定量)	流程評量, 流程分析, 量化的品質計劃	從預防問題發生	
	3 定型層 (Defined)	流程定義與制度化 (定性)	訓練, 技術性常規 (審查, 測試), 流程為中心 (標準化, 流程區分)	及早發現問題與矯正	
	2 規律層 (Repeatable)	流程依賴於個人 (直觀)	專案管理, 專案計劃, 組態管理, 軟體品質保證	專案最終的測試	
	1 起始層 (Initial)	(特別設置的 / 混亂)			

圖 3 能力成熟度模式 (CMM)

四、軟體流程改善與能力評定 (SPICE)

軟體流程改善與能力評定 (Software Process Improvement and Capability dEtermination, SPICE) 國際品保系統標準, 是由 ISO / IEC 的 JTC1 / SC7 在 1992 年 6 月簽署一份研究報告並成立 WG10, 且在 1993 年 1 月指派 WG10 從事 SPICE 國際品保系統標準的建立。同時也在 1993 年 6 月成立 SPICE 專案組織來從事推動該國際標準的相關工作。細部而言, 整個 SPICE 國際標準共有九個部份所組成, 其架構如圖 4 所示, 內容摘要則簡述如下:

第一部份: 它是 SPICE 國際標準的簡介, 並描述各部份如何適當組合與提供選擇和使用的指導綱要。它並解釋這標準與應用於進行評估、建構和選擇支援工具與延伸流程建構所應包含的要求。

第二部份: 它是定義高階層的基礎活動, 這些對軟體工程是相當重要, 且它是依流程能力水準的提昇而定。

第三部份: 定義進行評估時的架構, 並且設定評比、評分與描繪流程能力的基礎。

第四部份: 提供以團體基礎 (Team-based) 之軟體流程評估的指導綱

要。這綱要足以應用所有組織，並可進行使用不同方法、技巧與不同工具支援的評估。

第五部份：定義支援評估者進行評估時，建構設備的架構元素。亦提供取得者或設計者選擇與使用不同型態評估設備的指導綱要。

第六部份：描述進行流程評估時，相關評估人員的資格、教育、訓練與經驗。亦描述驗證資格與確認教育、訓練與經驗的機制。

第七部份：描述針對流程改善時，如何定義評估的輸入與運用評估的結果。這部份綱要亦包含在不同狀況下，應用流程改善的範例。

第八部份：描述針對流程能力評定時，如何定義評估的輸入與運用評估的結果。它指出在直接與複雜狀況下的流程能力評定，像未來的能力。進行流程能力評定的指導綱要除了提供組織評定自己的能力外，亦可讓取得者評定供應者的能力。

第九部份：SPICE 國際標準所特別定義的字彙。

參、軟體品保系統之架構

由第貳節的幾種品保模式之探討，雖然都是協助軟體業建立軟體品保系統的標準，大致上仍可分做三大類：ISO 9000 與 Tick IT 體系為強制性之標準，軟體業者可分為符合其標準或不符合二大類。CMM 則分得較細，將業者分做五大類，使其有一改善之方向。SPICE 國際標準則提出軟體流程評估的架構，可提供軟體組織運用在規畫、管理、監控、控制與改進軟體的取得、供應、發展、操作、展開與支援。並可協助軟體組織達到下列目的：(1) 讓組織針對流程改善時，瞭解自己流程的狀態；(2) 讓組織針對特定需求，評定自己流程的適合性；(3) 讓組織針對特定合約，評定其它組織流程的適用性。故若以 TQM (Total Quality Management, 全面品質管理) 不斷改善的作法而言 (Zultner, 1993)，顯然 SPICE 標準是最值得軟體業者所採用之架構。以下將對此架構中流程評估部分再作說明，並補足其不足之處。

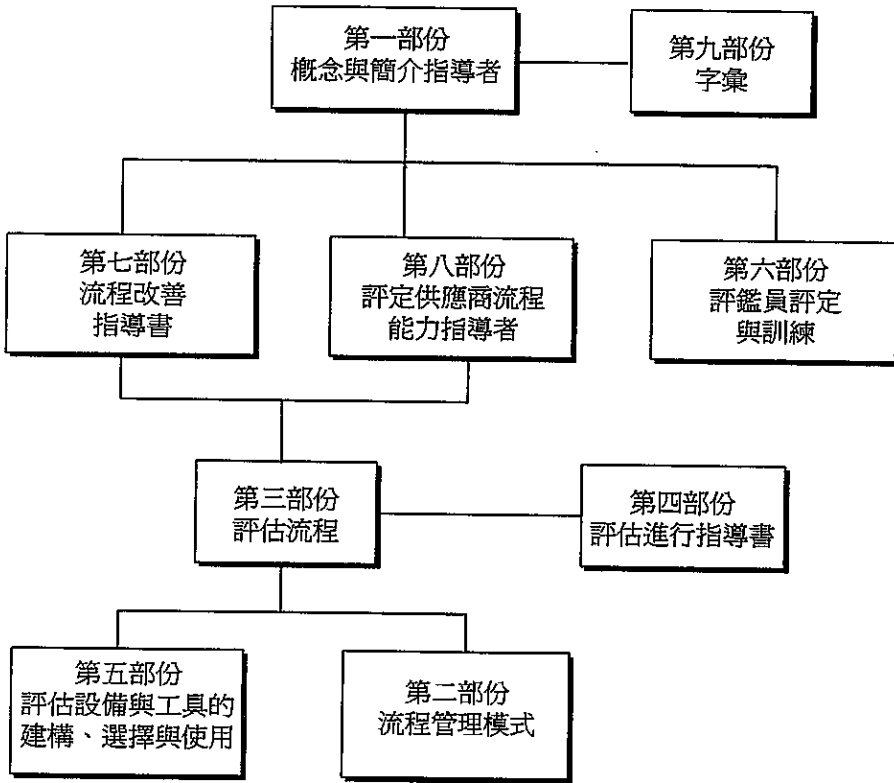


圖 4 SPICE 國際標準架構圖

一、SPICE 流程評估架構

SPICE 流程評估與流程改善、能力評定的整體架構，如圖 5 所示 (SPICE, 1996a)。在流程改善的內容中，流程評估可使組織根據流程之能力，提供一個具體實施的方法，並可依組織業務需要，分析評估結果來確認流程的強勢、弱勢與風險。更進一步可評估流程是否有效達成目標，與確認品質不佳或逾期或成本超支的原因。這些都是可提供管理者決定改善流程的先後順序。而流程能力評定是有關分析流程預定之能力與目標流程能力，來確認使用此流程去處理專案之風險。其中預定能力是建構在相關的事先流程評估結果，或是針對此預定能力所做流程評估結果的基礎上。

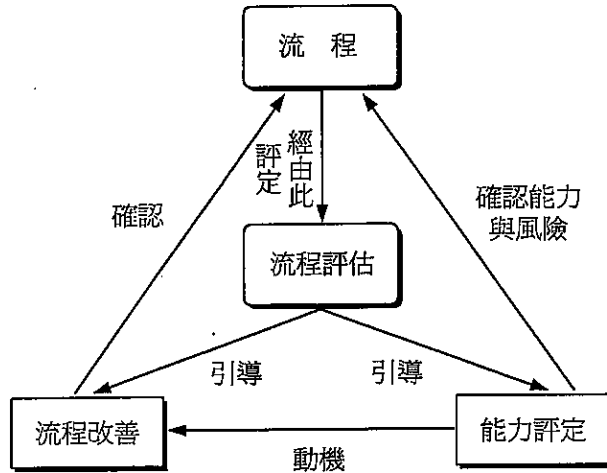


圖 5 SPICE 流程評估架構圖

在進行流程改善與流程評估時，必須針對軟體流程進行衡量，才能確認需要改善的流程之範圍並監控流程改善的績效。SPICE 流程衡量的架構，如圖 6 所示。其執行步驟與內容分述如下 (SPICE, 1996b)：

1. **組織的需求**：軟體開發組織的需求，例如軟體準時開發完成、開發成本不超支、產生的錯誤少等。
2. **軟體流程目標**：經由軟體開發組織的需要，衍生出關鍵軟體流程的目標。定義出軟體流程的目標後，可由軟體衡量來評估達成目標的程度。
3. **軟體流程度量**：數量化衡量軟體流程達成目標的程度。同時，軟體流程的度量必須為有效的衡量。
4. **軟體流程目標值**：設定軟體流程度量的目標值。
5. **軟體流程改善措施**：藉著改善措施的規畫與執行，來改進所有或部份的軟體流程。一個軟體流程改善措施也許會有助於一個以上軟體流程目標的達成。

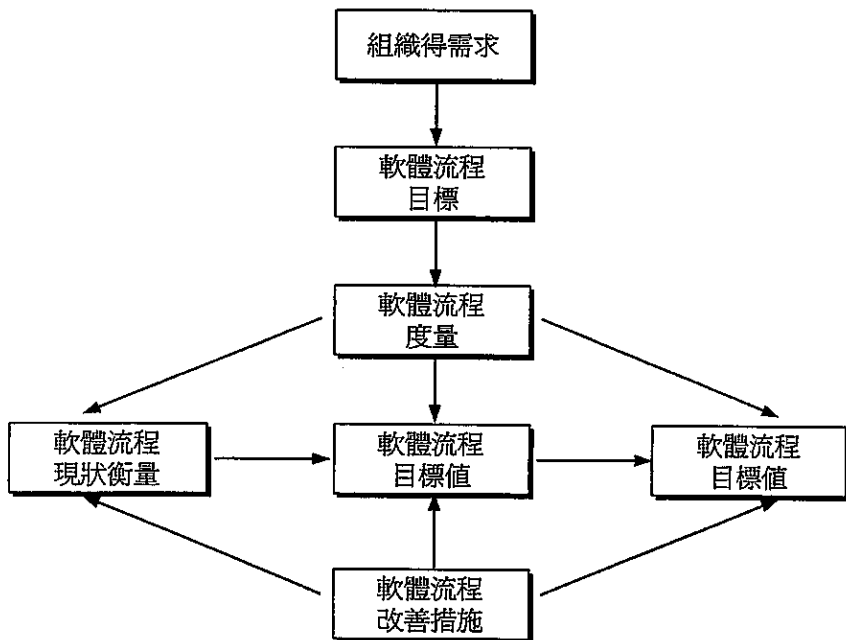


圖 6 SPICE 流程衡量架構

- 6. 軟體流程現狀衡量：在未執行改善措施時，軟體流程度量的值。
- 7. 軟體流程結果：在執行改善措施後，軟體流程度量的值，確認改善措施的成效。

二、架構之延伸

在上述架構中，對於如何數量化軟體流程之能力，及如何進行流程（製程，process）能力分析在標準內並沒有一具體辦法可循。所謂的製程能力分析是在探討流程達成目標的程度與準度，在流程為穩定的情況下，例如軟體開發人員有受過充分的教育訓練或有小型軟體專案的開發經驗，然後設定要進行衡量之特性的規格，並收集該特性的樣本資料，即可進行製程能力分析。在軟體業中，進行一個軟體開發專案時，每一個工作或軟體功能開發，並不是像製造業一樣，完全是相同的產品。但是它都是由相同的一群人來開發、撰寫程式、審查、測試等。因此可視為同一個流程，開發相似的產品，仍然可運用統計製程管制方法來監控、衡量軟體開發流程 (Pyzdek, 1993)。

在傳統製造業中，「製程能力 (Process Capability)」是指製程達到顧客要求之規格的能力。在製程為穩定狀態與製程為常態分配的情況下，製程能

力指標的定義如下 (Sullivan, 1985) :

$$C_p = \frac{USL - LSL}{6\sigma}$$

即規格寬度與自然容差之比。其中，USL 為規格上限，LSL 為規格下限， 6σ 為製程的自然容差。

由於製程的標準差 (σ) 未知，製程能力指標 (C_p) 的估計式為 (Kane, 1986) :

$$\hat{C}_p = \frac{USL - LSL}{6s}$$

其中 s 為樣本標準差。

由以上的定義，可知 C_p 只反映出自然容差與規格寬度的關係，在製程平均值與目標值不一致時， C_p 指標並無法顯示出完整的訊息。而另一個製程能力指標 C_{pk} ，可反映出製程平均值偏離目標值的程度，其定義如下 (Sullivan, 1985) :

$$C_{pu} = \frac{USL - \mu}{3\sigma}$$

$$C_{pl} = \frac{\mu - LSL}{3\sigma}$$

$$C_{pk} = \min \{ C_{pu}, C_{pl} \}$$

其中 μ 為製程平均。

由於製程的平均值 (μ) 與標準差 (σ) 未知，製程能力指標 (C_{pk}) 的估計式如下 (Kane, 1986)。

$$\hat{C}_{pu} = \frac{USL - \bar{x}}{3s}$$

$$\hat{C}_{pl} = \frac{\bar{x} - LSL}{3s}$$

$$\hat{C}_{pk} = \min \left\{ \frac{USL - \bar{x}}{3s}, \frac{\bar{x} - LSL}{3s} \right\}$$

其中 \bar{x} 為樣本平均數， s 為樣本標準差。

實際上，經由 C_{pk} 值計算出來之後，我們可以瞭解該特性達成目標的程度，與有多少機率會落在規格界限之外，如表 1 所示。

表 1 C_{pk} 值與不良率的對應表

C_{pk}	(落在規格界限外之次數) ／(每一百萬次機會)	C_{pk}	(落在規格界限外之次數) ／(每一百萬次機會)
0.20	274253	1.33	33.000
0.40	115070	1.40	13.400
0.60	35930	1.50	3.400
0.80	8196	1.60	0.793
1.00	1350	1.70	0.710
1.10	483	1.80	0.033
1.20	159	1.90	0.006
1.30	48	2.00	0.001

本研究建議結合 SPICE 流程衡量的架構，在軟體開發的過程中，建立製程能力指標(C_p 、 C_{pk})的運用，以衡量軟體開發流程，其架構如圖 7 所示，是以軟體功能為衡量的基礎，設定此軟體流程的目標，並進行實際地衡量與監控，以得到軟體流程衡量的結果，並以之為基礎來改善流程。

運用範例如表 2 所示，此例為某軟體開發組織為了能以具競爭性的價格取得軟體專案，於是衍生出軟體流程的目標為必須要準確地估計。而組織要評估的度量之一為設計階段成本估計準度之 C_{pk} 。據此設定此軟體開發流程的目標，並進行實際衡量與監控，以評估軟體開發之製程能力，本例中設定軟體流程目標值為 $C_{pk} > 1$ ，而實際評估之結果 C_{pk} 僅為 0.82，代表製程能力不足或流程不穩定，前者應該進行流程的改善，後者則應立即進行原因之分析，若屬特殊原因造成流程不穩，應立即進行特殊原因之消除；若屬共同原因，則應進行流程之改善。而在進行原因消除與流程改善之後，應再持續進行流程的監控與管制，以確保流程的品質水準。

表 2 軟體流程衡量範例

組織目標	軟體流程目標	軟體流程度量	軟體流程目標值	軟體流程結果
以有利的競爭價格取得合約	準確地估計	設計階段成本估計準度的 C_{pk} 值	$C_{pk} > 1$	$C_{pk} = 0.82$

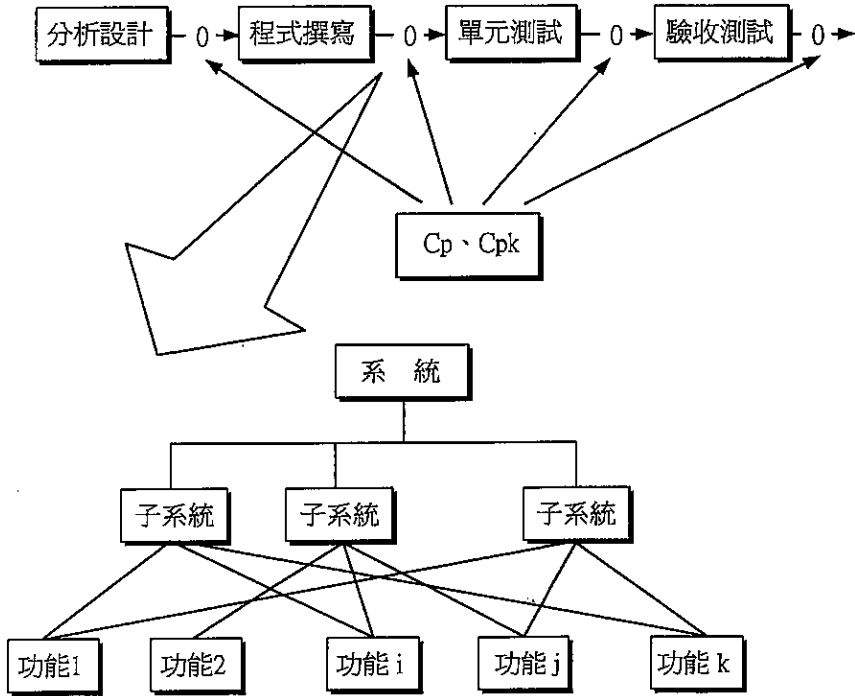


圖 7 軟體開發流程衡量架構

肆、個案研究

一、軟體產業現況

資訊工業多年來一直是政府發展工業的推動重點之一。經濟部工業局為達到公元 2002 年台灣製造業總產值 3200 億美元的發展目標，正全力推動「十大新興工業」的發展，希望到 2002 年時的十大新興工業總產值，可以由 1992 年的 293 億美元成長至 942 億美元，而其中光是資訊工業一項便佔了 330 億元，佔所有十大新興工業總產值的 35%強，因此資訊工業對於我國未來經濟的發展扮演非常重要的角色。而在 330 億元的產值目標中，資訊軟體即佔有 80 億美元。

根據經濟部產業技術資訊服務推廣計劃 (ITIS) 與資策會資訊市場情報中心 (MIC) 的資料顯示，目前我國軟體產業發展現況如表 3 所示，由表中可以看出軟體產業是一個高成長且深具發展潛力的產業，而最新公佈的資料顯示，1997 至 1998 年我國軟體外銷市場成長幅度為 37%，達到近新台幣 70

億元，其中幅度最大者為日本市場成長 80%，其次為北美市場成長 70%，兩者合計佔外銷總額的 77%，大陸與香港市場則成長 37%，至於東南亞市場則由於受到亞洲金融風暴的影響，衰退將近達 80%，歐洲市場亦衰退近 20%。

表 3 我國軟體產業發展現況 (1997)

單位：新台幣元

年度 項目	1992	1996	1997*	1992-1997
總營業額	263 億	531 億	645 億	19%
外銷額	16 億	43 億	60 億	30%
主要廠商	400 家	500 家	520 家	
人力	20,000 人	24000 人	25000 人	5%
總資本額	100 億	120 億	150 億	9%
硬體加值軟體**	200 億	500 億	600 億	25%
軟體人力	8,000 人	18000 人	21000 人	
企業開發軟體**	250 億	400 億	450 億	13%
軟體人力	30,000 人	42000 人	44000 人	

* 1997 為全年預測值，**為估算值

由資策會 MIC 調查我國軟體市場發展現況顯示，各區隔中以套裝軟體部份所佔比例最大，金額為 196.6 億新台幣，其次為系統整合部份，金額為 100.6 億台幣，再其次為轉鑰系統、網路服務、專業服務、處理服務等。其中又以網路服務業成長達到 31% 為最高，其次為套裝軟體約成長 17%，處理服務業因為延伸自 1995 年大型標案的後續，成長也有 11%。而其餘如系統整合、轉鑰系統、專業服務區隔，因缺乏大型標案的支撐，成長均低於 5%，詳如表 4 所示。

與硬體產業比較，我國軟體產業相對發展基礎薄弱，以套裝軟體市場為例，約有 60% (118 億台幣) 的市場實際上為國外大廠如微軟公司、蓮花公司等掌控。相對我國套裝軟體廠商在國內市場發展空間有限，如何突破限制尋求利基市場，是廠商重要課題。

表 4 1996 年我國軟體市場規模

單位：百萬新台幣

區 隔	1995	1996	95-96 成長率	1997(預估)	96-97 成長率
套裝軟體	16,807	19,664	17%	23,597	20%
系統整合	9,826	10,059	2%	11,059	10%
轉鑰系統	6,478	6,737	4%	7,411	10%
網路系統	4,499	5,881	31%	7,939	35%
專業服務	4,691	4,903	5%	5,903	20%
處理服務	1,467	1,622	11%	2,621	62%
合 計	43,768	48,866	12%	58,530	20%

爲了確保我國資訊工業之均衡發展，維持產業長期競爭力，工業局「軟體工業五年發展計畫（簡稱軟五）」的推動執行，軟五計畫成立之目的在於配合經濟部「十大新興工業發展策略及措施」項目下面的「軟體工業發展策略與措施」及「發展台灣成爲亞太製造中心推動計畫」細項下的「推動高科技高附加價值工業」中有關於軟體工業之發展政策，延續第一期「軟體工業五年發展推動計畫」以產品開發輔導爲重心的策略，運用有限的資源，重點培養軟體公司成功的明星案例，以吸引人才及資金的投入，加速帶動軟體工業的發展，使我國軟體工業在未來數年內除正常的自然成長率之外，能有更高的額外成長率，以達成公元 2005 年的發展目標。這中間並希望在 2002 年時，能使我國的軟體工業達到總營業額 2,000 億元新台幣（其中外銷 500 億元新台幣）的階段性目標。

爲達成此目標，如何系統化以提昇軟體公司品質系統，便成爲一重要之指標，藉由品質系統的建立，才能在激烈的競爭中，以最低成本，提供一致性高且品質優良的軟體或資訊系統。故目前一重點的推動方向，即爲盡早讓形象良好、具指標性的軟體公司能夠建立品保系統，致力獲得 ISO 9000 之驗證。

二、個案背景

爲了瞭解本研究所建立的製程能力衡量架構應用於實務上的可行性與遭遇的困難，因此本研究選取一個案公司作爲研究對象。此個案公司爲國內某軟體開發的機構，爲提昇軟體品質，該公司已取得 ISO 9000 的驗證。

此個案公司提供 A 公司委託其開發的存量管制與採購系統專案，給予本

研究作個案探討。

A 公司鑑於企業競爭日趨激烈，爲了有效控制物料、零件的存貨與採購作業狀況，因此委託個案公司開發存量管制與採購系統，來協助該公司追蹤物料、零件的採購與庫存狀況，以降低庫存成本，提昇企業的競爭力。A 公司的存量管制與採購系統，包括存量管制、修理零件、採購管理、基本資料維護與系統管理等五個子系統。其中採購管理又包括外購作業與內購作業，其系統架構如圖 8 所示。

基於商業機密，個案公司只提供 A 公司外購作業子系統的資料，供本研究作探討。外購作業子系統，主要是處理 A 公司國外採購相關的事務，涵蓋整個外購作業的流程，分別爲外購申請、狀況處理、狀況稽催、外購運況、外購驗收、外購接收、外購帳單、外購理賠、審核資訊、檢討報表與查詢，以及檔案維護與傳遞等 11 個功能。

三、資料分析

個案公司開發 A 公司外購作業子系統的流程，如圖 9 所示。個案公司提供本研究 A 公司外購作業子系統，開發過程中撰寫的程式行數、分析設計的估計與實際人力、程式撰寫的估計與實際人力、單元測試的估計與實際人力、驗收測試的估計與實際人力，以及單元與驗收測試的錯誤個數等資料。外購作業子系統的 11 個功能的各項資料，可整理如表 5 所示。

由於個案公司對於軟體開發流程品質衡量與監控之要求是平均每人-小時至少須撰寫 20 行程式，以及各個階段人力控制估計的準度誤差不得超過 $\pm 15\%$ 。其中「程式撰寫階段」的生產力特性是屬於單邊規格。因此個案公司在開發 A 公司外購作業子系統過程中，要監控的品質特性爲「程式撰寫階段」的生產力，也就是平均每人-小時撰寫的程式行數，以及四個階段人力控制估計的準度，也就是(人力實際值／人力與各個階段人力控制估計的準度，如表 6 所示。

經由表 6 的整理，可得開發外購作業子系統 11 個功能「程式撰寫階段」生產力與各個階段人力控制估計準度的樣本資料。經由適合度 K-S 檢定，發現這五個衡量特性的樣本資料均呈現常態分佈，如表 7 整理所示。在個案公司只有程式撰寫階段的生產力與分析設計人力估計的準度兩特性，符合組織的要求。經由本研究的製程能力衡量的結果發現，「程式撰寫階段」

的生產力，達成組織要求目標的程度不佳。同時，各階段的人力估計準度，達成組織要求目標的程度也不佳。其中程式撰寫、單元測試與驗收測試等階段之人力估計準度的 C_{pk} 值均為「負」的，顯示這些特性的平均值，落在組織要求的上下界限外，表示達成目標的能力極差，有必要立即進行改善措施。就整體而言，人力估計準度在開發此子系統的過程中表現不佳。

四、討 論

個案公司雖有提供此外購作業子系統，在單元測試與驗收測試階段的錯誤個數，但是「錯誤個數」是屬於計數型的特性，而 C_p 與 C_{pk} 的衡量是適用於計量型的特性，因此本研究不針對此特性進行製程能力分析與探討。所以，軟體開發組織在進行製程能力分析時，應先確認監控的特性是否為計量型，以避免誤用製程能力指標而產生錯誤的資訊。

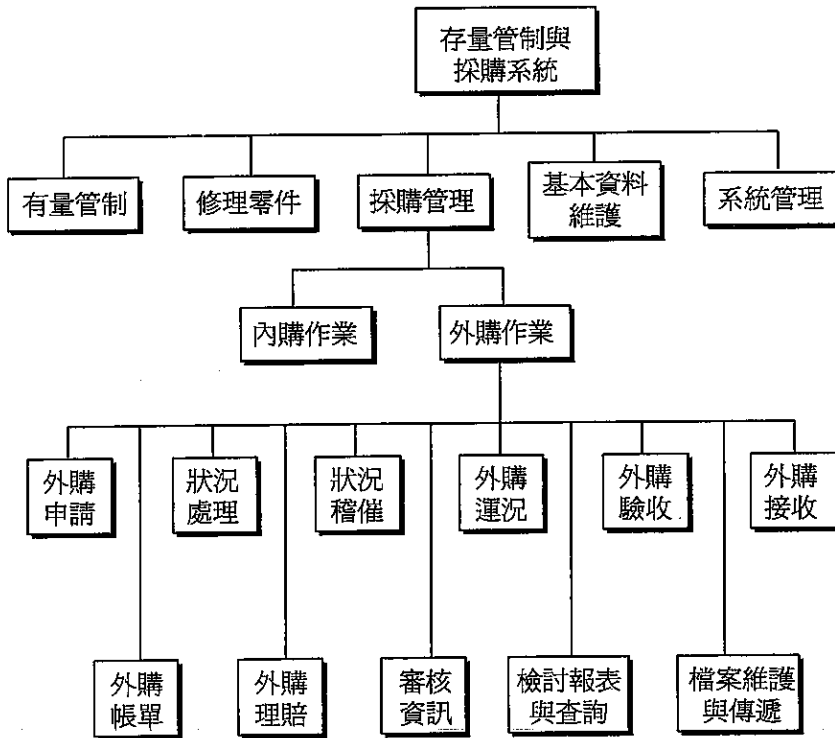


圖 8 A 公司存量管制與採購系統架構

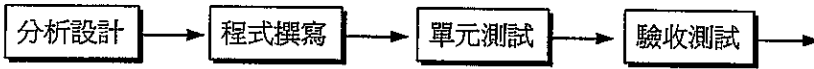


圖 9 A 公司外購作業子系統的開發流程

表 5 A 公司外購作業子系統 11 個功能的各項原始資料整理
(人力單位：人-小時，實際 / 估計)

序號	程式功能	程式行數	單元測試錯誤個數	驗收測試錯誤個數	分析設計人力	程式撰寫人力	單元測試人力	驗收測試人力
	外購作業子系統	24621	66	26	578/509	886/668	663/499	911/664
1	外購申請	1302	2	1	38/29	50/36	24/27	33/36
2	狀況處理	5155	15	7	143/114	216/142	125/110	177/138
3	狀況稽催	986	2	1	16/20	26/26	24/20	32/26
4	外購運況	806	2	1	19/32	44/40	39/34	46/46
5	外購驗收	865	5	3	39/32	64/40	41/28	66/38
6	外購接收	1278	1	2	20/22	37/30	29/20	44/32
7	外購帳單	2528	7	1	88/78	109/96	76/66	130/82
8	外購理賠	1097	4	1	15/16	32/26	22/18	27/26
9	審核資訊	2029	7	1	44/40	69/56	45/42	74/56
10	檢討報表與查詢	6641	10	6	96/94	150/128	154/102	232/140
11	檔案維護與傳遞	1934	11	2	60/32	89/48	54/32	50/44

在製造業中，製程能力指標 C_p 與 C_{pk} ，代表產品達到規格要求的能力，同時隱含產品的不良率。而在軟體業中，顯示軟體開發流程達到組織要求的能力，隱含在一百萬次的機會中，落在組織要求界限外的次數。兩者代表的意義不全然相同。

針對此個案各階段之「人力估計準度」進行衡量的結果，發現在各階段的 C_p 與 C_{pk} 值均偏低，顯示人力控制的能力不佳，導致花費的人力增加，使得開發成本大為提高，甚至可能會成為賠本的狀況。而且，開發的時間會產生連鎖性的延遲，導致無法準時將系統開發完成交予顧客。

軟體開發流程的 C_p 與 C_{pk} 值偏低，顯示流程的變異很大，流程不穩定，可能隱含軟體開發人員能力不佳或專案管理不佳，專案的方向沒有作充分的溝通，範圍沒有作完整的定義，導致決策品質不佳，專案變動性高。針對這些問題，軟體開發組織有必要加以重視，並應進行改善措施。若要提昇軟體開發流程能力，有必要更落實 ISO 9000 軟體品保或 SPICE 的要求，並隨時運用統計製程管制方法進行監控，發現異常應立即予以改善，以免產生連鎖性的影響。

表 6 A 公司外購作業子系統 11 個功能之衡量特性樣本資料的整理

序號	程 式 功 能	程式撰寫之生產力	分析設計人力估計準度	程式撰寫人力估計準度	單元測試人力估計準度	驗收測試人力估計準度
外購作業子系統		27.79	1.14	1.3263	1.2685	1.3720
1	外購申請	26.0	1.31	1.3889	0.8889	0.9167
2	狀況處理	23.9	1.25	1.5211	1.1364	1.2826
3	狀況稽催	37.9	0.80	1.0000	1.2000	1.2308
4	外購運況	18.3	0.59	1.1000	1.1471	1.0000
5	外購驗收	13.5	1.22	1.6000	1.4643	1.7368
6	外購接收	34.5	0.91	1.2333	1.4500	1.3750
7	外購帳單	23.2	1.13	1.1354	1.1515	1.5854
8	外購理賠	34.3	0.94	1.2308	1.2222	1.0385
9	審核資訊	29.4	1.10	1.2321	1.0714	1.3214
10	檢討報表與查詢	44.3	1.02	1.1719	1.5098	1.6571
11	檔案維護與傳遞	21.7	1.88	1.8542	1.6875	1.1364

表 7 A 公司外購作業子系統開發，各特性的衡量結果整理

	程式撰寫階段之生產力	分析設計人力估計準度	程式撰寫人力估計準度	單元測試人力估計準度	驗收測試人力估計準度
個案公司的衡量	27.79	1.14	1.3263	1.2685	1.3720
適合度檢定 (K-S Test $\alpha = 0.05$)	0.128 < 0.391 Normal	0.177 < 0.391 Normal	0.263 < 0.391 Normal	0.212 < 0.391 Normal	0.127 < 0.391 Normal
C_p	0.2883	0.1502	0.1978	0.2150	0.1834
C_{pk}	0.2883	0.0455	-0.2179	-0.1667	-0.1813
(落在界限外次數) / (一百萬次機會)	203660	678130	783750	737920	766330

伍、結論與未來研究方向

一、結 論

本研究除了探討幾種軟體品質系統外，並結合 SPICE 流程衡量架構，建立一軟體開發流程製程能力衡量之架構。此外，研究中選取一個實際軟體開發專案，進行軟體開發流程製程能力分析，以驗證本架構的可行性。本研究尚有以下幾點結論：

1. 軟體開發過程中，關鍵流程特性有必要加以監控，以免導致開發時間延遲、成本超支等問題。本研究發展之製程能力指標衡量架構，同時考量組織的目標與流程變異程度，能顯示出流程達成目標的程度，似較傳統的衡量方法為佳。
2. 軟體開發組織運用製程能力分析來監控關鍵流程特性時，有必要先確認此特性是否為計量型，以免誤用製程能力指標 (C_p 、 C_{pk})。

二、未來研究方向

由於本研究為相關文獻中首次探討如何將製程能力指標 C_p 與 C_{pk} 導入軟體開發流程的衡量，雖以一案例說明其實用性，但仍有許多值得深入探討之處。未來的研究可就下列方向來進行。

1. 軟體開發的過程中，有些關鍵流程會受多重特性的影響，未來可就軟體開發流程中，多重特性製程能力指標的運用，作一深入探討。
2. 本研究未考量衡量特性的樣本資料間是否有相關性存在，未來可針對樣本資料間有相關性存在時，探討該如何進行製程能力分析，以及它對製程能力指標的影響。
3. 未來可就本研究發展之製程能力衡量架構，將它開發成一個軟體工具。藉由電腦化的收集、擷取樣本資料，並計算與分析各個衡量特性的製程能力，以簡化流程衡量的工作與降低人力的運用。

陸、問題討論

- 一、本文中提及的四種軟體品保系統之間的差異與適用時機為何？
- 二、運用 C_{pk} 值量化軟體開發流程能力有何優缺點？
- 三、經由 K-S 適合度檢定，如果不符合常態分配時，應如何進行後續評估作業？
- 四、如果品質特性屬於計數型態時，應如何調整此軟體品保之系統？
- 五、個案公司以「程式撰寫階段生產力」與圖 9 中四階段的「人力控制估計準度」為監控與衡量軟體開發流程品質保證系統的品質特性，要應用在

其他軟體開發業者時，是否應該有其他的品質特性？

柒、致 謝

本研究特別要感謝梁中平先生熱心提供許多珍貴的資料，與兩位評審委員提出許多寶貴的意見，以及博士班研究生粘孝增協助整理資料，最後要感謝國科會提供部份之經費支持（計劃編號：NSC 85-2416-H-006-013），使本研究得以順利完成。

捌、參考文獻

- Johnson, K. 與陳秀梅，「ISO 9000 軟體品質體系 (Tick IT) 簡介」，品質管制月刊，1995年4月，頁34~36。
- 呂執中與李明章，「軟體品質保證之利器：ISO 9000-3」，品質管制月刊，1994年8月，頁55~62。
- BSI, "Quality Systems: Part 13. Guide to the Application of BS 5750: Part 1 to the Development, Supply and Maintenance of Software," 1991.
- Clapp, F., "Getting Started on Software Metrics," *IEEE Software*, 10(1), 1993, pp. 108-109, 117.
- Costello, R. J. and Liu, D.-B., "Metrics for Requirements Engineering," *Journal of Systems and Software*, 29(1), 1995, pp. 39~63.
- Freese, J. D and Konold, E., "Teamwork Pays off for Firm," *Quality Progress*, 27(5), 1994, pp. 53~55.
- Kane, V. E., "Process Capability Indices," *Journal of Quality Technology*, 18(1), 1986, pp. 41-52.
- Pyzdek, T., "Process Control for Short and Small Runs," *Quality Progress*, 26(4), 1993, pp. 51-60.
- SPICE, "Software Process Assessment - Part 1 : Concepts and Introductory Guide," *SPICE Management Board*, 1996a.
- SPICE, "Software Process Assessment - Part 7 : Guide for Use in Process Improvement," *SPICE Management Board*, 1996b.
- Sullivan, L. P., "Letters," *Quality Progress*, 18(4), 1985, pp. 7-8.
- Zultner, R., "The Deming Approach to Software Quality Engineering," *Quality Progress*, 1988, 21(11), pp. 58-64.

