

全面品質管理於軟體開發之應用

陳鴻基

清華大學科技管理研究所
hgchen@mx.nthu.edu.tw

李有仁

加州理工州立大學商學院資管組
eli@calpoly.edu

摘要

在「全面品質管理」的概念下，對產品品質有全新的看法和定義。事實上，自從工業革命以來，歐美國家的工業活動焦點，已逐漸改變成以產品為重心。所謂「只要能生產，必有購買者 (If I make it, someone will buy it)」的觀念，即普遍深植於歐美製造業者心中。七十年代石油危機後，儘管汽油價格的迅速上揚，汽車製造商依舊相信美國人會繼續購買美國汽車，仍繼續生產大而無效率的汽車，因而銷售量受到嚴重影響。同時，國外小型且更具效率的汽車陸續打入美國市場，更嚴重地侵蝕美國汽車製造商的市場佔有率。他們大多不願深入了解顧客真正的需求，造成後來持續性的銷售衰退和損失。那時，他們對品質的看法如下：

- (1) 生產力和品質是互相衝突的
- (2) 當必須保持生產力的話，改善品質往往會消耗組織額外的資源。因此，品質的改善常會犧牲生產力。
- (3) 品質被定義為符合規範或標準
- (4) 這樣的觀念容易使公司養成墨守成規的做法，而較不會注意到所採用「規範」、「標準」的不合理性、和不合時性。
- (5) 品質以不合乎規範的程度（比例）來衡量
- (6) 通常採用 6-Sigma 的方法，衡量每百萬工件中不良工件數來定義品質，如此的衡量，常常忽略了更重要「顧客的滿意度」的指標。
- (7) 品質的保證是必須靠密集式的檢驗來達成
- (8) 如此做法，需要消耗大量組織的資源。
- (9) 只要達成最低品質要求，即使產品有缺點也可接受
- (10) 這樣的前題下，認為顧客會付費並接受有“bug”且不完善的產品。
- (11) 品質是一項獨立性的作業且集中於生產線上來執行
- (12) 如此觀點，認為生產線團隊將會與品質單位合作融洽。
- (13) 品質的不良是因線上工作者的疏失或供應商的疏失所造成
- (14) 與供應商關係完全是靠成本考量或短期性的契約來維繫

以上品質觀點，在 1980 年代「全面品質管理」概念提出後，受到嚴重的挑戰而有重大的改變。「全面品質管理」在結合了不同的品質專家的看法後，才慢慢成型。這一概念發芽於二十年代，當 Walter Shewhart (1931, 1939) 在貝爾實驗室介紹利用統計上的控制機制來降低檢測與實驗的變異。後來五十年代時期，W. Edwards Deming (1981, 1982, 1986) 將此概念帶到了遠東地區並替戰後日本企業建立起組織層級的品質控制藍圖。Joseph Juran (1945, 1951) 跟著介紹「全面品質控制」於生產上的應用。而 Kaoru Ishikawa (1976, 1985) 在六十年代更進一步的提倡品質應該是每一位從業員工與上層管理人員的共同責

任。他介紹了品質圈的作法。Genichi Taguchi (1986, 1987)強調使用數量方法和實驗設計法來改善品質，並應用在產品設計上。在美國的方面，首先 Armand Feigenbaum (1951, 1957, 1961)在五十年代介紹了全面品質控制的觀念。而七十年代，由 Philip Crosby (1967, 1969, 1979)開始推行零缺點的標準。

經過多年探討，全面品質管理的目標，已從過去追求利潤最大化，成本最小化，和經濟規模等有形目標，轉變為提昇顧客滿意度，提昇品質和降低時程等有形與無形目標。本文將探討當代全面品質管的觀念與準則，剖析執行方法與步驟，討論 Deming 的管理機制，並將其應用到軟體開發中加以探討。

1. 全面品質管理

根據韋氏大字典的解釋，品質是優良的程度；一種獨特的屬性。簡言之，就是一產品在功用，可靠性及保養等各方面均達到優良程度及其它顧客需求的屬性。為了產出一優良的產品，就須在產品開發過程中植入全面品質管理概念。「全面」，表示企業內每一件事物的全盤性。包含每種製程，每個批次，每種資源，每次產出，以及人、時、地。美國品質控制學會(ASQC)認為，全面品質管理是一種透過滿足顧客而達到長期成功的管理方法。該方法是基於企業中所有成員的參與，以改善製程，產品，服務及本身的文化。Philip B. Crosby、W. Edwards Deming、Armand V. Feigenbaum、Kaoru Ishikawa 以及 J. M. Juran 等皆是大力提倡並教導企業勵行全面品質管理的大師。

此外，全面品質管理是哲學也是一套提供遵循的守則，同時也是企業追求持續性改善的基礎。全面品質管理運用數量方法及人力資源以改善企業內部所需的原料及服務，而且達到顧客需求的程度。全面品質管理是以嚴謹的方法，著重精益求精，因此針對基本的管理技巧，已改善項目的投資，以及技術性的工具加以整合。

2. 全面品質管理下的品質觀點

全面品質管理的運作，是由以顧客為中心，由顧客加以衡量。藉由製程持續性改善來提高產品或服務的品質，其重心是試圖達到整個企業內部的全部品質，而非僅產品本身。企業內部的改善，舉凡對零組件的設計改良或是制度的流程改善，將可改善企業及其最終產品的全面品質。此種看法迥異於傳統的觀點：

1. **生產力和品質並不是互相衝突的。**經由品質改善可提升生產力。因為較佳的生產製程可減少重製品，錯誤及浪費。因此可以提升生產力。
2. **品質的定義，在於符合顧客的需求。**一個產品的最終品質繫於滿足顧客的需求。因此我們在規劃產品設計階段時就應該考量顧客層面。Juran (1974)認為

品質的良窳，由使用者加以評斷是可行的。

3. **品質是透過顧客滿意度來衡量以及在製程與設計上持續性改善。**顧客喜歡購買不僅合乎其需求而且表現超乎水準的軟體。
4. **品質的達成仰賴於有效的產品設計及製程控制。**以往依賴產品的檢驗來完成品質的保證。但當不良品產生時，這種事後補救方法不能全然達到品質水準。所以，為了建立品質，必須要有良佳的產品設計以及流程控制。
5. **透過流程控制技巧來防止缺失。**如果企業可以持續的改善品質，完美無缺的流程理應是企業的目標。
6. **品質工作是產品生命周期各個階段的一部份。**隨便地進行生產卻沒有一個品質相關的計劃，產品或服務的品質不可能良好，更別奢談會符合顧客的需求。
7. **管理者須對品質負起責任。**唯有管理人員有權改變工作流程與條件，也唯有管理人員有相關知識，可以在整個產品生命週期中對各個攸關品質的功能加以協調。
8. **與供應商的長期關係是由品質考量來維繫。**供應商是一企業整體團隊中重要的一部份。管理者對品質負有責任，就應負責與供應商維繫由品質考量的長期關係。

3. 執行全面品質管理的原則

根據文獻探討，收集不同企業實施全面品質管理的經驗，以下列出成功實施全面品質管理的指導原則：

1. **品質是與眾人攸關之事。**品質是管理者和每位員工該關注的事。利用賦予員工獨立自主的授權能力，品質就能顯著改善。員工對於其負責的工作因具有優良表現產生歸屬感。汽車工業就授與其員工在生產線上因發現品質瑕疵品而隨時停工的權力。
2. **強調顧客的重要性。**每人必須專注於滿足內部和外部的需求和期望，不僅僅符合規範即可。培養顧客至上的生產趨勢，所有投入的改善努力皆為迎合最終產品的顧客或是內部顧客的需求和期望。整體生產流程上，每人負責自身處理的產品或在製品的品質改善，對緊接其下一個生產製程的員工負起品質改善之責，所以全面品質管理，每個人都有其顧客。
3. **品質必須內建於產品。**品質不是單指最後檢驗的後見之明。任一產品在前半部的設計規劃，就須牢記品質問題。它是從開始至最後都須注意的問題。品質工作不是指製程完成後挑選出良品，把不良品重做的想法。在全面品質管理下，不應該容許產生不良品。瑕疵品在任一製造流程就會被發現。因此，品質是內建於產品，是產品的一部分。在產品研究開發每一階段，研發者應須具這一觀念。
4. **全面品質管理要求管理者的信念及全程參與。**全面品質管理須在企業中由上而下的執行，若管理者對它沒有信念，是無法達成的。全面品質管理執行是

否成功，繫於管理者的堅定的領導。

5. **成功地執行全面品質管理需要持續性訓練。**持續性改善包含每個人須改善其執行負責的工作表現。每位員工須學習全面品質管理的原則及其執行的工具、技巧。並把員工的學習成果視為績效評估的一部份。
6. **領導能力替代口號及訓詞。**我們常在企業內部聽到類似「品質第一」等口號，例如福特汽車公司。但只說而不去執行，口號還是空談。當福特成功地推出 Taurus 新車專案時，他們依舊產出瑕疵車。福特所需的是品質改善的更佳領導力。
7. **長期強調生產力改善及可衡量的程序。**全面品質管理不是一夜之間就能執行。它是必需花費數年去執行的長期性程序。企業內部的文化改變，集中於持續性改善，且需要衡量工具加以比較改善前與改善後的不同。
8. **了解未開始改善前的目前流程。**我們必須了解目前的流程，才能知道能有多大的改善空間。為了要比較前後的改善程度，就要有一套衡量工具。
9. **強調交互職能及團隊工作。**一個交互職能的團隊，其在研究開發階段時就開始把企業內各種不同的部份加以結合。例如：軟體產品的研發設計團隊必須囊括財務部門、會計部門、行銷部門以及其它部門的成員，因為研發一項新產品不僅僅是研發部門人員的事。即使是配銷部門和維修部門的員工在產品研發階段都應投入。
10. **有效使用統計方法及品質控制工具。**使用統計品質控制及流程控制技巧，辨識發生超出控制限度的偏差原因，並採取行動消除此例外原因。常被使用的品質控制工具，例如：「Q7 工具」和「M7 工具」等，藉這些工具收集有用資料並力求改善。表 1 列出 Q7 工具和 M7 工具的大略，表 1 和表 2 各自描述 Q7 工具和 M7 工具的格式。Q7 工具是用來分析歷史資料，解決特定問題。大部分的問題在製造階段發生，且需要各個不同職能的人員共同討論和解決。在此情況下，無論是追求產品品質改善、成本減少、新產品研發或是政策修訂，M7 工具可派上用場。
11. **製程、產品及服務不斷地改善。**採行全面品質管理，追求不斷改善的企業文化才能成功。授權所有員工去影響整個企業，一起改善品質。一旦給員工權力，他們就會不斷地表現自我信念和一股慾望去改善公司。他們積極找尋方法改善份內職責及企業整體的品質表現，所以管理者也應該透過合宜的獎酬和肯定來改革其組織文化。
12. **讓每人有參與全面品質管理的誘因。**如果把全面品質管理視為一把火把，誘因則是這火把的燃油。全面品質管理的精神，大部份來自員工針對成本抑減的建議。經由一個跨部會的意見評估團隊評估這些建議，具有顯著貢獻的建議將被採行，提出此項建議的員工將受到肯定與金錢和名譽獎。例如：德州儀器半導體公司便會舉辦品質受獎儀式。

表 1：品質控制工具

品質七項工具(Q7 Tools)	管理七項工具(M7 Tools)
<ol style="list-style-type: none"> 1. 要因圖 (Cause-and-effect diagram)。也稱為「魚骨圖」。該圖可以辨識，探討，及呈現與問題有關的成因。 2. 檢查表 (Check sheet)。這份文件是用來對定期對環境的檢測後的結果作成表格。並且在完成後傳給生產程序的各個主要管制點，以擔負起缺失把關的任務。 3. 管制圖 (Control chart)。用來偵測變異的異常原因。該圖有中心線、上界線、與下界線，樣本的資料數值就直接在圖上描繪其對應的點，再利用管制圖來評估流程的狀態與趨勢。 4. 直方圖 (Histogram)。主要是以條狀圖來顯示資料的次數分配。管理人員利用該圖來判斷資料的中位數，偏態與峰態。 5. 圖形 (graphs)。有許多不同型式的圖表可以對評估人員提供協助。直線圖 (Line graphs) 可以用來說明一段時間內的變異。除此之外，還有柱狀圖 (Bar graphs)，圓形圖，圓餅圖用來顯示每類數值佔整體的比例。雷達圖則協助分析先前已評估過的資料，再以資料本身的測量軸重新分析。 6. 柏拉圖 (Pareto chart)。該圖將問題根據其原因與現象加以分類，並且利用柱狀圖以指定的次序，呈現問題在某個選定項目的相對重要性。 7. 散佈圖 (Scatter diagram)。又可以稱做 X-Y 圖。該圖可以在一個變數改變的情況下，顯示另一個變數的是否也受其影響，以驗證理論或預測。 	<ol style="list-style-type: none"> 1. 類似圖 (Affinity diagram)。這是腦力激盪下的產物。它是根據團隊而來，團隊中的每個成員寫下自己的主意，而後再依據主題，將這些成員所提供的意見重列於主題之下。 2. 箭形圖 (Arrow diagram)。又稱做流程圖。通常使用在 PERT (Program Evaluation and Review Techniques) and CPM (Critical Path Method)上。該圖利用網路的表達方式來顯示要實行計劃或完成程序的必要步驟。 3. 矩陣圖 (Matrix diagram)。該圖是用來澄清個不同因子間的關係。通常使用在品質需求到相對應的特質上，然後再到產品需求上。 4. 矩陣資料分析圖 (Matrix data analysis diagram)。這個圖是用來支援矩陣圖，特別是矩陣圖的詳細度不足時。這是管理七項工具中唯一的一項根據資料分析並且提供數值結果的工具。 5. 流程決策計劃圖 (Process Decision Program Chart, PDPC)。這個圖一般是由作業研究的團體使用。該圖本身是利層級方式來展現如何達到一個最佳的替代方案。類似於決策樹。 6. 關聯圖 (Relations diagram)。這個圖有另有一個名稱：原因模型圖。該圖釐清一個複雜情況下包含到許多交互關聯的因子間的相互關係。並且可以清楚的顯示因子間因果關係。 7. 樹狀圖 (Tree diagram)。該圖類似於功能分解圖。可以應用在顯示目標與方法間的相互關係以及流程與活動間的相互關係。

資料來源：摘錄自 Ishikawa [10], Imai [9], Brassard [4], and Walton [17]

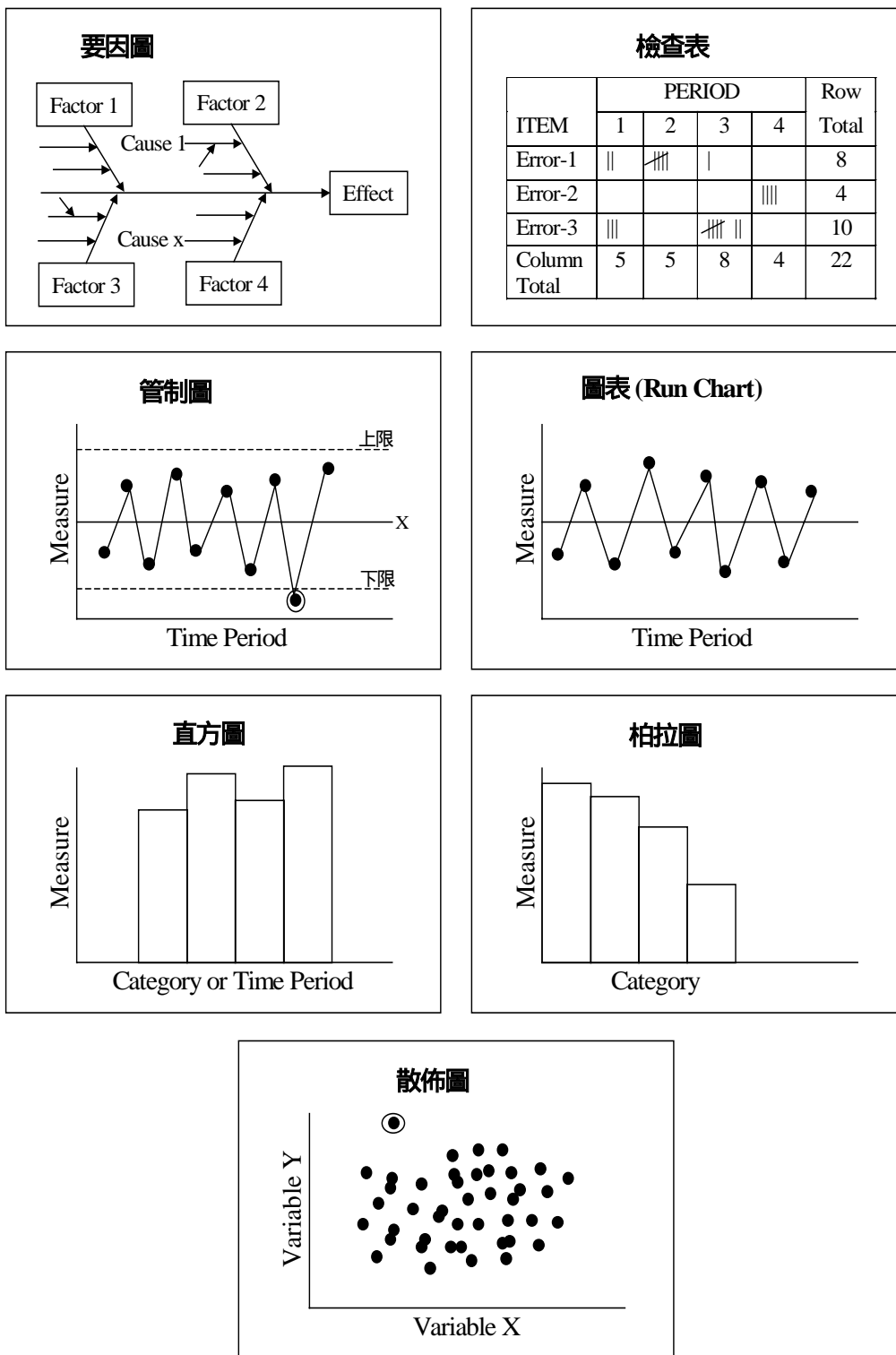
13. **資訊共享**。團隊合作是全面品質管理能否成功的關鍵。它仰賴與同一職能內的員工和其它職能部門的員工共同分享必要的資訊和專業知識。同時，也需要員工督促自我更加努力工作達成公司整體及個人目標。然而，應避免有些

非必要的資訊共享，例如：薪資給付額及紅利水準等，因為會造成反功能的後果。

14. **消除溝通障礙。**在 TQM 的文化下，管理幹部與工作人員之間以及部門與部門之間不應該有溝通障礙。工作人員如果有需要，可以輕易的找到想要詢問的管理幹部。員工所提出的建議應該實施，以消除溝通障礙。
15. **供應商必須執行全面品質管理的哲學。**若一產品的原料或配件具有瑕疵的問題，公司不可能產出具品質的產品。因此，供應商也需執行全面品質管理，才能確保供應商提供的原料符合本公司所要求的品質需求。類似及時生產制度 (JIT)，全面品質管理哲學倡導和供應商須有穩固的合作關係。公司須減少供應商的數目，並和同樣具全面品質管理哲學的供應商達成長期的生意允諾，這促使供應商會自我持續性品質改善，以確保其產品的品質。

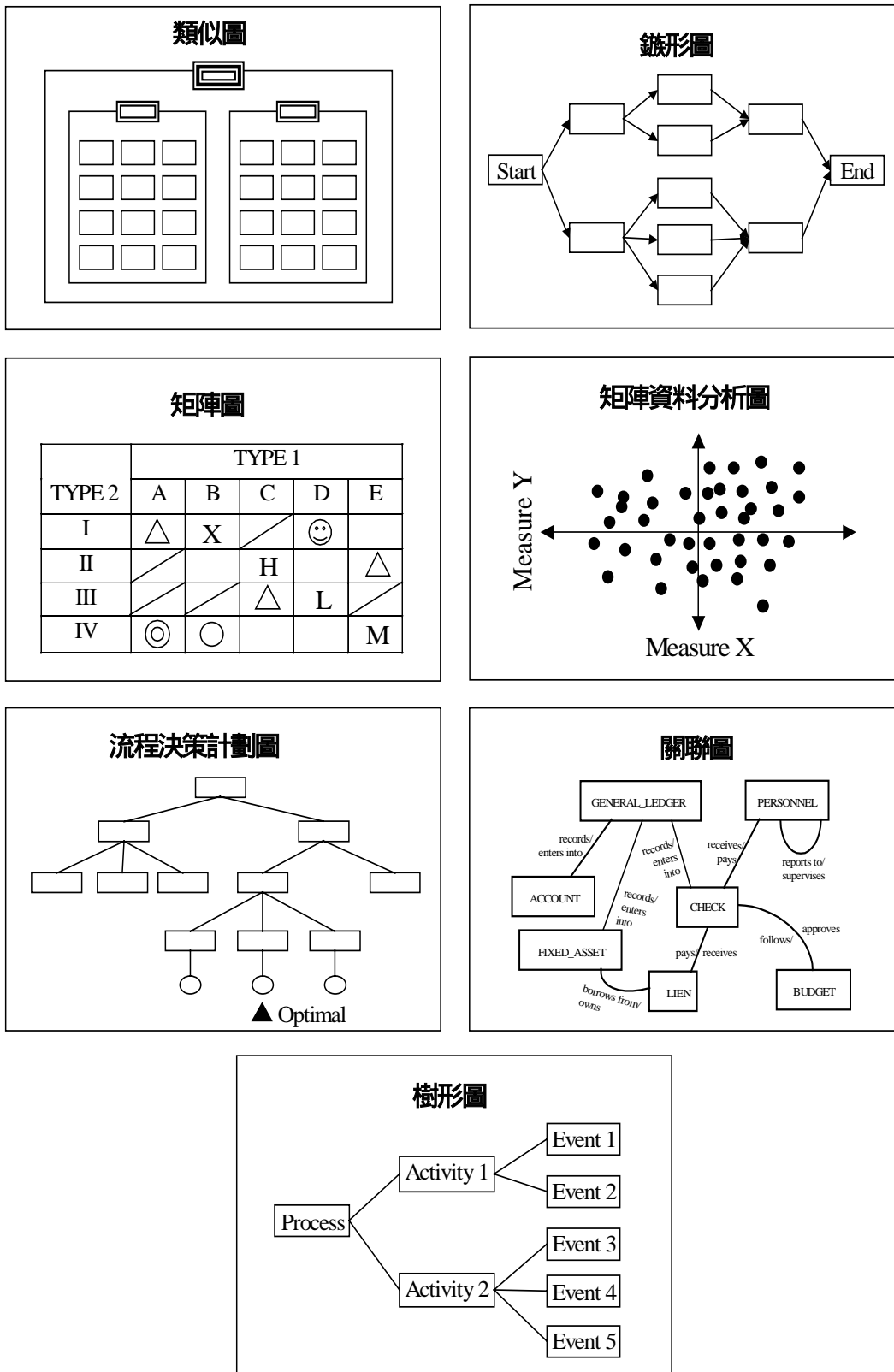
表 2：系統開發方法論各階段詳述
(SDM)

雛型開發生命週期階段	系統開發方法論	階段目標
需求分析	服務需求/計劃可行性評估	開始一項專案，並進行成本/效益分析與可行性研究。
	系統需求定義	定義專案範圍，分析目前的系統，定義資訊需求，資料屬性，與系統目標。
設計	系統設計 替代方案	定義與評估替代性的系統設計，並且開始規劃專案的時程。
	系統外部 規格	確定資料流，使用者/系統介面，系統控制以及手動支援程序。
	系統內部 規格	確定處理邏輯，檔案結構，模式介面，以及系統架構。
建立雛型	程式開發	使用程式語言將程式的內部規格轉換為程式碼。
測試雛型	測試	驗證與確認經由 SDLC 開發的系統。
先期/完全發行	轉換	轉換資料型態與程序到新的系統上。
	安裝	為新系統安裝硬體與軟體，並利用新系統開始生產。
後續工作	後續實施 重新檢視/維護	監控新系統的品質與表現。



資料來源：摘錄自 Ishikawa [10] and Brassard [4]

圖 1：品質七項工具



資料來源：摘錄自 Brassard [4] and Walton [17]

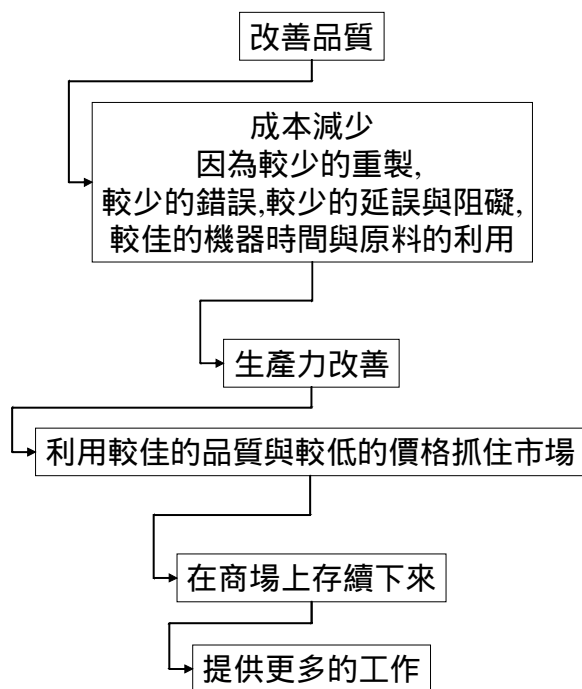
圖 2：管理七項工具

4. 戴明管理方法

Walter Shewhart 被推崇為統計品質控制制度的始祖,然而 W. Edwards Deming 是第一位導入全面品質管理觀點的學者。戴明的主要觀點,總結在其十四要點上：

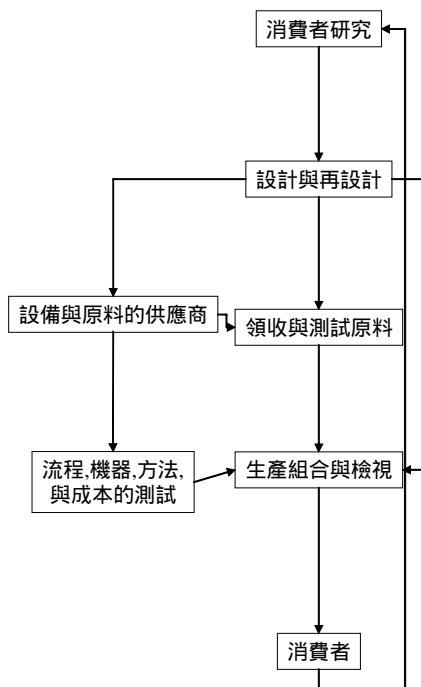
1. 創造產品與服務改善的恆久目的
2. 採納全面品質的新哲學
3. 停止依靠大批量的檢驗來達成品質
4. 廢除“價格低者得”的做法
5. 不斷地及永不間斷地改善生產及服務系統
6. 建立在職培訓方法
7. 建立領導能力
8. 驅走工作不安全的恐懼
9. 打破部門之間或員工之間的圍牆
10. 加諸於員工身上之口號、訓詞等目標應予廢除
11. 取消工作標準及數量化的定額
12. 任何導致員工失去工作尊嚴的因素必須消除
13. 建立嚴謹的教育及訓練計劃
14. 協助每位員工去完成此項轉變

隱藏在這十四要點背後,是戴明管理方式的哲學。此哲學可透過戴明(1986)提出的二個圖作個小結：連鎖反應圖及永無終點圖 [Deming, 1986]。戴明主張在連鎖反應圖裡頭,不僅公司本身可以受惠,甚至擴及整個社會。因為由該圖可以減少公司內部的成本支出,而且向社會大眾提供更多的工作機會。換句話,與其每個人搶一塊餅,但是餅並沒有變大;改善品質所做的事就是將餅變大,讓每個人可以分享的量變多。戴明更進一步強調,永無終點圖所表示的意思是顧客的眼中自有其對品質的看法。品質是為了供顧客所需。針對顧客的研究,必須了解顧客使用該項產品的目的為何。為了滿足顧客的需求,必須持續改善產品的品質與流程,同時也需要求供應商有相同的回應。戴明的想法是創新遠比變革的立意來的好;品質的達成是藉由持續性的改善組織,而不是突然對組織做大幅度的變動。此外,全面品質有賴於公司與其供應商間的工作小組。如果你無法與一個供應商合作,就別打另一個供應商的算盤。最後,公司儘可能的針對任何一個項目都只鎖定一個供應商。



資料來源：節錄自 Walton [17].

圖 3：戴明的連鎖反應圖



資料來源：摘錄自 Walton [17].

圖 4：戴明永不止息的流程圖

5. 針對軟體流程改善的全面品質管理

以上所述的全面品質管理的理論，可以應用到所有開發中的流程，不論其流程是針對產品或是軟體。

6. 產品開發生命週期 (PDLC)

產品開發生命週期是一個有系統、有次序的方法，藉此管理產品開發相關的活動。該方法主要是遵循 Herbert A. Simon 所提出的問題解決步驟：情報收集、設計方案、決策制定、與結果反思 [Simon, 1977]。新產品的開發開始於需求分析的階段。在這個階段期間，主要是收集顧客需求，並加以分析、評估，以便開發出產品規格。根據顧客需求與產品規格，可以在設計階段，開發出產品設計藍圖。這些藍圖中包括了製造設計規格，以及原料的領料單。根據這些藍圖，可以建造出產品的雛型，並加以測試，評估整個雛型的品質。如果測試未通過，本次失敗的案例可作為分析與辨識之用。找出失誤的地方是在於雛型建造的流程，或是產品設計流程。但是最糟的部份是以上兩個流程都沒出錯，錯的是在進行需求分析階段，不當的可行性分析所訂定的產品規格。在這種情形下，整個產品開發計劃可能必須取消。在圖五的虛線部份顯示出失敗的原因可以依序地反推回錯誤流程所在處。一旦，產品雛型通過了所有測試後，雛型中最佳的一個將被選來進行先期測試發行（一種有限規模的發行，以測試市場反應），或是正式發行的版本。上任何一種發行，都需要開發出一條專門針對該項產品的生產線。如果先期測試的結果並不令人滿意，銷售資訊就被反推回需求分析步驟，同時產品需求將再進行重新評估。相反的，如果先期測試成功，整個產品就可以準備大量生產以進入全面發行。再進入接下來的步驟。如果接下來的報告顯示出產品銷售成功，而且有利潤，這個訊號代表可以直接再回到全面發行的階段，繼續大量生產。假如，報告結果剛好相反，就必須再進行需求分析，重新再跑一次產品開發生命週期的各個階段。

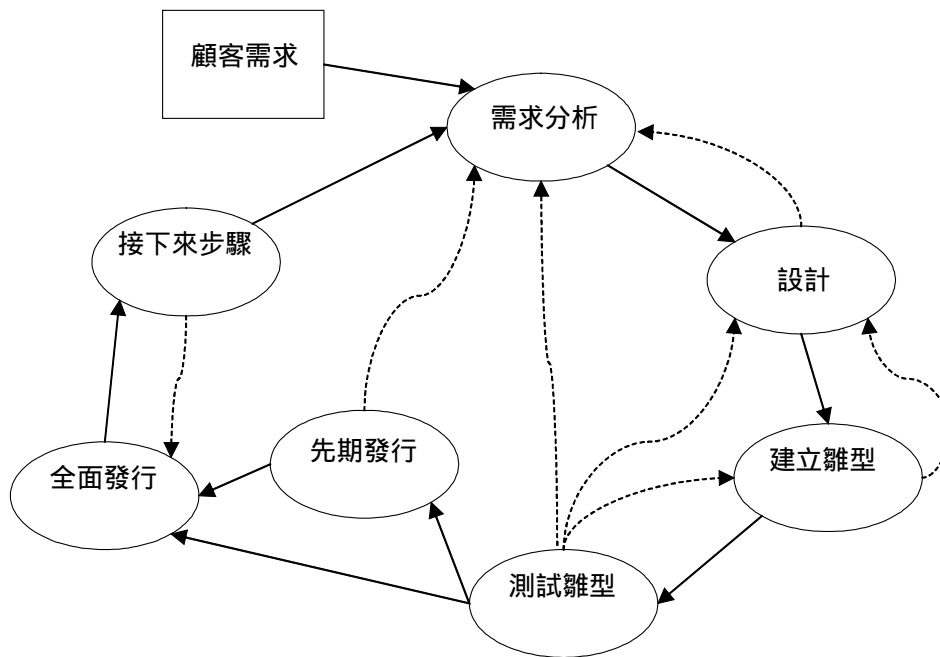


圖 5：產品發展生命週期

7. 系統開發生命週期 (SDLC)

系統開發生命週期類似產品開發生命週期。通常包含了以下各個步驟：分析，分析，設計，實做，與支援 [Whitten and Bentley, 1998]。依照所需要的系統結構，系統開發生命週期可以遵循結構化的開發方法論 (SDM) [Davis and Olson, 1985; Li, 1990]，快速雛型方法論 (RPM) [Boehm, 1984; Lantz, 1986; Naumann and Jenkins, 1982]，或是螺旋形開發與強化方法 (SDEM) [Boehm, 1988]。SDM 基本上應用在一個需求定義清楚，處理及報告均結構良好，而且期望長期穩定的系統。SDM 流程的詳細分析列在表 2。使用 SDM 的方法論時，不允許在流程中來回重覆各個階段。理論上，SDM 本身可以稱為「瀑布式」的流程。相反地，RPM 允許而且也鼓勵流程中各個階段的來回重覆。RPM 使用高層次的開發工具以便快速的製作出可使用者操作的雛型，再由使用者的經驗中獲取經驗。然後，整個雛型再依據使用者反應加以改良。這種流程適合於具有特殊能力的系統，但是其需求不明確或是模糊不清；或者，生命週期短，而且需求隨時變動的系統。最後，SDEM 將 RPM 與 SDM 兩個流程結合，以縮短採用 SDM 流程所需的開發時間。近來，物件導向開發方法論開始取代傳統的 SDM 流程，而成為系統開發的新典範。

傳統上，要確保軟體產品的品質是在軟體開發過程中實施軟體品質保證技術（SQA）。但是今天，單單 SQA 並不能達成顧客所要求的軟體品質。必須將 TQM 的方法應用至整個開發軟體的組織，而不是只有流程本身而已。這種方式已被稱為軟體全面品質保證（STQM）。STQM 從傳統的品質觀點到組織哲學做整個文化上的改變。也就是，這個改變包含了組織各個層面的品質改善。SQA 提供了一個確保品質的方法論，但是 STQM 提供了一個架構以持續改善品質。

Masaaki Imai [1986] 認為獲利的最高指導原則，可以透過提升品質，縮減成本，縮短排程來達成。藉由在軟體開發的各個階段貫徹 TQM 持續改善策略，不管之前產品品質如何，都可以達到其前所未有的水準。以新的觀念與策略不斷地挑戰現狀，協助持續提供軟體工業更佳的产品。許多軟體公司正嘗試將這種文化帶入其內部，甚至藉由強化基層員工的能力來協助進行改善的工作。

8. 應用戴明的十四點原則於軟體開發

戴明管理方式十四點提供 TQM 實施方式的指引。這十四點可以應用在管理軟體開發過程上。以下討論主要是依據最後一節所談的系統開發生命週期架構：

1. **針對產品與服務的改善，建立永久不變的目標。**傳統的軟體開發流程在整個開發完成的系統轉交給負責支援的團體，而且整個系統上線開始生產後就結束。而在 TQM 的文化中，開發團隊是沒有所謂終點線的。也許，在不同的階段，關注的焦點會轉移至另一個專案。但是，開發團隊該應該對其所交付的系統負起責任，而非由支援的團隊負責。在生產時，任何有關品質的問題都應該告知開發團隊。管理人員必須 [Zultner, 1988]：
 - 在軟體開發過程中，於每一個階段建立操作性定義。
 - 定義何謂「針對顧客的服務」。
 - 針對下年度，以及五年之內的開發、維護、服務等項目，訂立標準。
 - 定義內部與外部的顧客。
 - 找出以較少的時間與資源，卻可以提供較佳系統與服務的方法。
 - 投資於獲得較佳軟體開發的工具與技術上。
2. **採用全面品質的新哲學。**品質是組織中眾人之事。管理人員也是品質團隊中的一部份。在 TQM 的文化中，品質第一而且組織中人人都必須參與。企業由上到下的管理人員必都須擁抱 TQM 的概念，而且要讓軟體開發團隊的所有成員清楚了解，其對 TQM 的支持。
3. **減少以大量檢查的手段以達成目標的方式。**品質是內建而非外加。在撰寫程式碼時即應防止錯誤的產生而不是為了去除錯誤才重新撰寫程式碼。檢查或是測試均無法防止錯誤的發生；唯有經驗與知識才能達到防範於未然。管理

人員必須建立持續地改善軟體開發流程的計劃，例如：工作訓練與工作動機計劃。

4. **結束單單以價格標籤來獎勵商業的習性。**今天，許多軟體組織將其專案外包給其它的合約商。所以很重要的一點就是不應僅以價格來決定合約。品質比價格的差異性更加重要。長期的品質低落最終會付出極高的代價。儘量與一些忠實而且程式碼的品質值得信賴的供應商建立長期的夥伴關係。
5. **持續性，長久性地改善生產與服務的系統。**透過新式的方法論，典範，標準，策略，技術，工具，政策，及程序以持續改善系統開發流程。組織要達成以上所提的方式，必須不斷地透過管理資訊系統方面加以追蹤，以找出最佳的方法，也就是所謂的學習型組織。而組織中的個體必須藉由提升或擴展個人技能的方式來迎合組織的要求。
6. **在職訓練的機制。**為了將品質內建於產品中，開發團隊必須具有適當的經驗與知識。在職訓練的計劃是獲取這些經驗與知識的有效的方法。廣義而言，所有的 MIS 成員必須知道他們的工作完成後產生的結果，以及其工作方式。管理人員必須在分配員工到軟體專案之前，對員工的技巧層級進行評估。不同的技巧層次在一個專案中可以扮演不同的角色，以及承擔不同的責任。
7. **領導統御的機制。**管理人員著重在領導而非懲罰。管理人員的工作在於幫助 MIS 人員工作上有較佳的表現，以及創造較佳的工作環境。專案管理人員必須接受人際關係與分析技巧的訓練。而且對於統計程序控管要確實的了解。管理人員應該了解在任何軟體開發團隊中，即使團隊的半數成員的表現在控管範圍內，最後的結果仍是低於平均值。管理人員必須把焦點擺在表現脫離控管的成員身上。
8. **消除對工作不安的恐懼。**員工在放心地發問、提建議、甚至暴露其缺點以尋求協助以前，必須有安全感。長期雇用政策應可以消除面對工作產生不安的恐懼感。而且，任何 MIS 員工其表現在統計控制以下的，應該提供該員再訓練，或職務調動。但是，如果該員工持續拒絕他人或其上司的幫助，解雇可能是最後的手段。
9. **打破部門或是員工地區屬性的藩籬。**軟體開發要求來自使用者與 IS 人員的合作。我們要謹記，溝通歧異已成為 MIS 實施失敗的因素之一。而且，今天企業的資訊系統專案極有可能包含不同的功能部門，而且需要來自資料庫處理，主從伺服器運算，以及網路安裝的專業知識。因此，跨部門的開放式的溝通，並且具備不同學門的一般性知識對一個成功的系統實施有其必要性。要使團隊的成員達到此要求，必須透過適當的教育與訓練，以改善成員的知識所得。
10. **取消勸誡勞工的口號。**標語沒有辦法建立高品質的系統。MIS 管理人員不應該訂下不可能的目標或時程表，或是不切實際的生產力水準。相反的，管理人員在回應建議與幫助員工改善品質的同時，應該公布員工進步的幅度。讓員工提出自己的口號與標語。

11. **取消數值配額與工作標準。**配額（如工作量），目標（如排程）以及工作標準（如單位時間所完成的數量），位址數目等，並未與品質畫上等號。一個急就章與缺乏一致性的軟體開發計劃不僅無法完成，連後續的服務也不用談。應由專案成員建立成員本身的目標。專案經理應該從旁協助成員減少工作的重覆性，錯誤率，以及相關的虛耗事項。每個員工的工作方向是持續性的改善，而非隨意訂下的短期目標 [Zultner, 1988]。
12. **拆除員工自尊心的藩籬。**所有員工都有其動機，想要建立優良的產品。然而，優秀的員工也必須有原料，工具，方法，時程的配合，才能建立優秀的產品。原料短缺，工具不堪使用，方法無效，時程拖延，都是有損員工自尊心，必須清除。提供軟體開發團隊，用團隊的代號或是隊名為產品命名的機會，以彰顯團隊的貢獻。
13. **建立組織成員重新教育與重新訓練的有效機制。**為了某項特殊工作，以在職訓練，提供員工學習某項技術的計劃，雖然有效但是效果緩慢。在今日的 MIS 領域，短短期間內，類型相同的工作所需的新技術變化極快。管理員必須有充足的預算，以進行充裕的教育訓練，讓組織成員得以進行自我改造。在 TQM 的文化中，所有員工必須知道足夠的統計方法以了解變異的本質，並能管理變異中的特殊成因。透過員工訓練，使員工獲得必備統計方法的支持必須制度化。
14. **促使組織內成員徹底執行工作，以完成企業轉換。**TQM 的轉換是組織內部眾人之事。每個人都是顧客。問自己，什麼人將會收到你工作後的成果。每個人都必須藉由自我審視找出自己所面的顧客對象為何，以精確定義自我工作。每個人都隸屬於一個團隊，都是位居計劃 - 執行 - 檢查 - 行動（plan-do-check-act）循環中的重要要環節。而且，管理必須要可行。唯有管理人員才能確實改變影響個人工作表現甚深的企業文化與環境。管理人員必須了解管理人員的意義與執行方向。同時，管理人員必須知錯必改。而且應負責向組織成員解釋為何改變是必須的，以及改變所影響的對象包括組織中的所有人。所以很明顯地，眾人要知道何去何從，一定要了解戴明的十四點原則 [Walton, 1986]。

9. 結論與建議

全面品質管理不只是工作上的哲學，同時也是工作人員的倫理。TQM 來自於許多品質改善的大師多年教學的智慧結晶。TQM 幫助許多公司改善產品與流程的品質，而且更進一步提升生產力與獲利能力。任何要搭上 TQM 列車的軟體組織，在計劃實行 TQM 之前，一定要讓組織內部的所有員工擁抱 TQM 的哲學與方法。也就是，不管員工層級，都必須了解（即使是訓練也可以）TQM，而且要把 TQM 的概念與工具內部化。為了增加成功的機率，TQM 的實行計劃目標必須使整個組織的員工都受益，而且必須由高層管理人員開始，向下擴展至基

層的管理員與員工。講的具體些，其目標就是針對員工，透過工作環境，工作方式，工作報酬，工作關係的改善，來改善員工的工作生活品質，並且提供員工專業發展的機會。也唯有這個目標，我們才能獲取員工合作，成功的實行 TQM。

要注意的是天下沒有白吃的午餐，一旦實行 TQM 的概念與方法，你就必須持續不斷地改善你的產品與流程。你必須一直問自己：「我要如何做與做什麼，才能在下次有更好的表現」。計劃 - 執行 - 檢查 - 行動的巨輪是沒有終點線的。每個與產品的價值鏈有關的個人都是不可或缺的，門必須持續的用心來演好他們所扮演的角色。否則，整個價值鏈會分崩離析，TQM 的流程也很快就會瓦解。所以，整個 TQM 實行所牽連的人員都決定了 TQM 成功與否。因此，為了容易管理起見，總數不要超過三十人。典型的軟體開發計劃，其團隊總人數通常不超過三十人，以適合 TQM 的實施。對於一個成員數較大的組織，我們強烈建議使用模組化的方法來實行 TQM。千萬不要嘗試想要一次將整個複雜的組織轉換成符合 TQM 的組織。專家經驗告訴我們，此種方式成功的機會微乎其微。

參考文獻

- [1] Bemowski, K. (ed.), "The Quality Glossary," *Quality Progress*, February 1992, p. 28.
- [2] Boar, B. H., *Application Prototyping: A Requirements Definition Strategy for the 80s*, New York: Wiley-Interscience, 1984.
- [3] Boehm, B. W., "A Spiral Model of Software Development and Enhancement," *Computer*, Vol. 21, No. 5, May 1988, pp. 61-72.
- [4] Brassard, M., *The Memory Jogger Plus*, Massachusetts: GOAL/QPC, 1989.
- [5] Davis, G. B., and M. H. Olson, *Management Information Systems: Conceptual Foundations, Structure, and Development*, New York: McGraw-Hill, 1985.
- [6] Deming, W.E., *Out of the Crisis*, MIT Center for Advanced Engineering Study, Cambridge, Mass., 1986.
- [7] Department of Defense, *Total Quality Management: A Guide for Implementation*, DOD 5000.51-G, Feb. 15, 1990 (Final Draft).
- [8] Department of Defense, *TQM Guide, Vol. 1, Key Features of the DOD Implementation*, DOD 5000. SIG. Final Draft, 15 FEB 1991, p. 2.
- [9] Imai, Masaaki, *Kaizen: The Key to Japan's Competitive Success*, New York: McGraw-Hill, 1986.
- [10] Ishikawa, K., *Guide to Quality Control*, 2nd rev. ed., Tokyo: Asian Productivity Organization, 1982.
- [11] Lantz, K. E., *The Prototyping Methodology*, Englewood Cliffs, NJ: Prentice-Hall, 1986.
- [12] Li, E.Y., "Software Testing in A System Development Process: A Life Cycle Perspective," *Journal of Systems Management*, Vol. 41, No. 8, August 1990, pp. 23-31
- [13] Moore, W. S., "Singing the Same 'Total Quality' Song," *National Defense*, March 1990, pp. 29-32.

- [14] Naumann, J. D., and A. M. Jenkins, "Prototyping: The New Paradigm for Systems Development," *MIS Quarterly*, Vol. 6, No. 3, September 1982, pp. 29-44.
- [15] Simon, H. A., *The New Science of Management Decision*, Revised Edition, Englewood Cliffs, N.J.: Prentice-Hall, 1977.
- [16] Strickland, J., "Total Quality Management," *Army Research, Development & Acquisition Bulletin*, March-April 1988, p. 1.
- [17] Walton, M., *The Deming Management Method*, New York: Putnam, 1986.
- [18] Whitten, J. L. and Bentley, L. D., *Systems Analysis and Design Methods*, 4th ed., New York: McGraw-Hill, 1998.
- [19] Zultner, R., "The Deming Approach to Software Quality Engineering," *Quality Progress*, November 1988, pp. 58-64.