

國立交通大學

高階主管管理學程碩士班

碩士論文

分散式軟體工廠之軟體開發模式



Distributed Software Factory Model for
Software Development

研究生：陳裕昌

指導教授：李榮貴 教授

中華民國九十四年六月

分散式軟體工廠之軟體開發模式

Distributed Software Factory Model for
Software Development

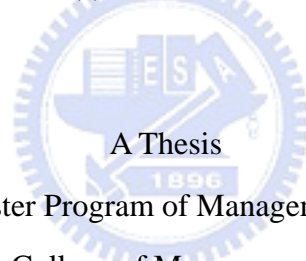
研究生：陳裕昌

Student: Yu-Chang Chen

指導教授：李榮貴 教授

Advisor: Zong-Kwei Li

國立交通大學
高階主管管理學程碩士班
碩士論文



Submitted to Master Program of Management for Executives

College of Management

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Executive Master

of

Business Administration

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年六月

分散式軟體工廠之軟體開發模式

學生：陳裕昌

指導教授：李榮貴 教授

國立交通大學 高階主管管理碩士學程碩士班

摘要

目前台灣資訊軟體業者大多數規模不大，很難在國際舞台與國際大廠競逐，台灣硬體製造業發展至今 22 年，累積相當豐碩的國際成就，反觀軟體產業，40 年下來產值還非常小。台灣軟體業的問題在於沒有標準開發方式，無法承接大型專案，國際競爭力不足的問題，源自於軟體產業規模不大，無法像製造業一樣形成供應鏈。生產力與品質是軟體成功的關鍵，而成本控制與時程掌握是公司經營績效的主要因素。因為缺乏標準化製程，許多軟體開發便倚重人力，不同的人員素質產生不同的品質，因此品質不一，成本無法控制，時程也缺乏可預測性。本研究從製造業生產的角度來看軟體開發問題，以文獻探討和參考個案為主，個案以國內上市的系統整合電腦公司所屬的軟體工廠為研究對象，個案公司分別在大陸、台灣都建置軟體工廠，承接軟體代工設計與編碼，是一個標準分散式軟體生產模式。透過個案公司來了解目前建置軟體工廠的做法與方向。

關鍵字：軟體工廠、軟體開發流程、軟體元件再使用

Distributed Software Factory Model for Software Development

Student : Yu-Change Chen Advisor : Dr. Zong-Kwei Li

Master Program of Management for Executives
National Chiao Tung University

ABSTRACT

Most of the software companies in Taiwan are small and medium-sized scale. It is difficult for them to compete with other companies in the international market. The hardware manufacturing industry accumulates rather plenteous international achievement in the last 22 years. However, software industry production value is still very small during the past 40 years. Many of software companies are lack of standard development process. They can't handle large project. The international competition ability is poor because of the small scale. They are not able to build up supply chain like hardware manufacturing industry. The successful key factors for software companies are productivity and quality. And good cost control, project schedule are the main factor of the company management results. Many software are developed depending on manpower because they don't have standard manufacturing process. Therefore quality is different; the cost is uncontrollable; the schedule is also unpredictable. The purpose of this research is in order to solve software development problem from the viewpoint of manufacturing industry with paper study and individual case study. In this case study, we choice a famous system integrated company who has software factory in both Taiwan and Mainland china. This factory takes full charge in software design and coding process. It is a distributed software factory development model. We clear the process of software and direction of building the software factory from this case.

誌 謝

EMBA 學習過程是一段難忘的經驗，在經歷一段漫長的職場生涯，對工作內容逐漸感覺單調乏味時，由於 黃種智學長的推薦，讓我重新體驗校園生活，給工作帶來新的刺激。EMBA 同學們來自不同的產業，常常有不同的互補效果，開拓了視野也豐富了人生經驗。

論文撰寫期間承蒙 李榮貴教授指導，給我很大研究的空間，也讓我在時程安排上非常自由，口試委員 蔡志弘教授與 張盛鴻教授提供建言及指導，在此一併致謝。凌群電腦公司同仁－曾鎮崑 協理、黃鎮洲 協理、林靜蘭 經理。數美科技 婁植山 總經理，可取科技黃叔敏 執行顧問，若比鄰科技 丁豔光 協理等都提供了寶貴資訊，對本研究貢獻良多。

最後，感謝老婆與女兒的支持與鼓勵，讓我不得不努力向前，你們是我最大的動力來源。

陳裕昌 謹誌

民國九十四年六月二十九日

目 錄

中文摘要	i
英文摘要	ii
誌謝	iii
目錄	iv
表目錄	v
圖目錄	vi
一、	緒論.....	1
1.1	研究背景與動機.....	1
1.2	研究目的.....	7
1.3	研究方法.....	8
二、	相關文獻探討.....	9
2.1	軟體工廠 (software factory)	9
2.2	軟體再使用元件(software reuse component)	12
三、	軟體工廠生產模式.....	15
3.1	軟體工廠解決策略.....	15
3.2	軟體工廠組成.....	16
3.3	從訂單到派工的製程規劃.....	18
3.4	生產中心的規劃.....	20
3.5	軟體模版設計與標準化工具.....	23
四、	軟體工廠個案.....	24
4.1	個案公司背景.....	24
4.2	分散式軟體工廠組織流程改造.....	24
4.3	協同作業平台的工具支援.....	29
4.4	建立績效指標.....	38
五、	結論.....	41
5.1	軟體工廠成功關鍵因素.....	41
5.2	未來研究方向.....	41
參考文獻	43
個人簡歷	50

表目錄

表 1 軟體工廠和製造工廠的差異.....	15
表 2 程式等級與標準工時.....	18
表 3 量化指標.....	39
表 4 指標推算說明.....	39
表 5 非量化績效指標.....	39



圖目錄

圖 1	2003-2007 我國資訊軟體市場規模.....	1
圖 2	資訊軟體廠商營收分佈百分比	2
圖 3	資訊軟體廠商員工人數分佈百分比	2
圖 4	各國平均勞動成本與技能評比	5
圖 5	專案調查分析(2000)	6
圖 6	專案調查分析(1994-2000).....	6
圖 7	軟體開發面臨的問題	7
圖 8	NEC Advanced software Factory	11
圖 9	Toshiba Fuchu SF..	12
圖 10	經驗庫軟體開發模式.....	13
圖 11	HP 軟體實驗室 - 產業別應用元件	14
圖 12	元件組合軟體開發	14
圖 13	軟體工廠組成.....	17
圖 14	軟體開發各階段產出	17
圖 15	軟體工廠部門職掌.....	20
圖 16	軟體工廠上游規格.....	21
圖 17	生產中心規劃.....	21
圖 18	軟體再使用元件階層.....	21

圖 19	生產工具整合.....	23
圖 20	依產業別分工組織	26
圖 21	組織再造-依功能別分組	26
圖 22	功能別組織	27
圖 23	軟體工廠組織	27
圖 24	協同作業圖	28
圖 25	軟體供應鏈	28
圖 26	SYSCOM Online 架構圖	30
圖 27	Project online 關聯圖	34
圖 28	Project online 功能	34
圖 29	Project online 的文件管理	36
圖 30	SDMaker 設計畫面	37
圖 31	SDMaker 程式碼產生器	38

一. 緒論

1.1 研究背景與動機

資訊軟體業係指從事套裝軟體、系統整合、轉鑰系統及專業服務等四項業務的公司。根據資策會市場情報中心的統計 [1]，目前台灣資訊軟體業者共有 1,200 家，2003 年產值為新台幣 1,500 億元，2004 年產值為 1,576 億元，平均成長率為 5.1%，預估到 2007 年時產值可達新台幣 1,907 億元（圖 1 所示）。然由於廠商家數過多，絕大多數規模都不大 [2]，公司員工在 100 人以下的比例為 79.6%，營收規模在新台幣 5,000 萬元以下的比例有 54.6%，平均產值在 500 萬以下者佔 74.2%（圖 2 所示）。因為公司規模不大，很難在國際舞台與國際大廠競逐，成為台灣軟體產業發展一大隱憂。

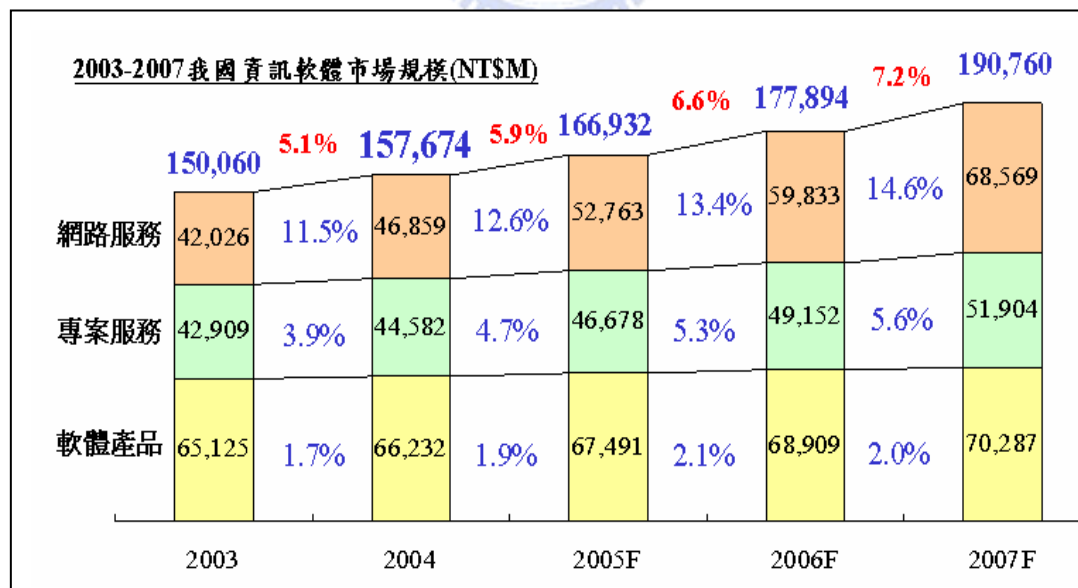


圖 1：2003-2007 我國資訊軟體市場規模 (單位:百萬新台幣)

資料來源:資策會 MIC 情報中心, 2005/01

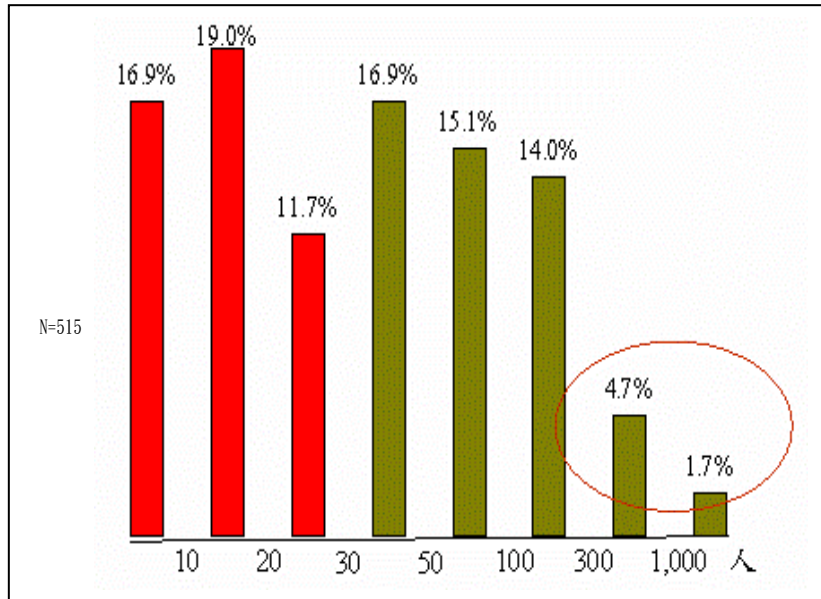


圖 2：資訊軟體廠商營收分佈百分比
資料來源:資策會 MIC 情報中心，2002/12

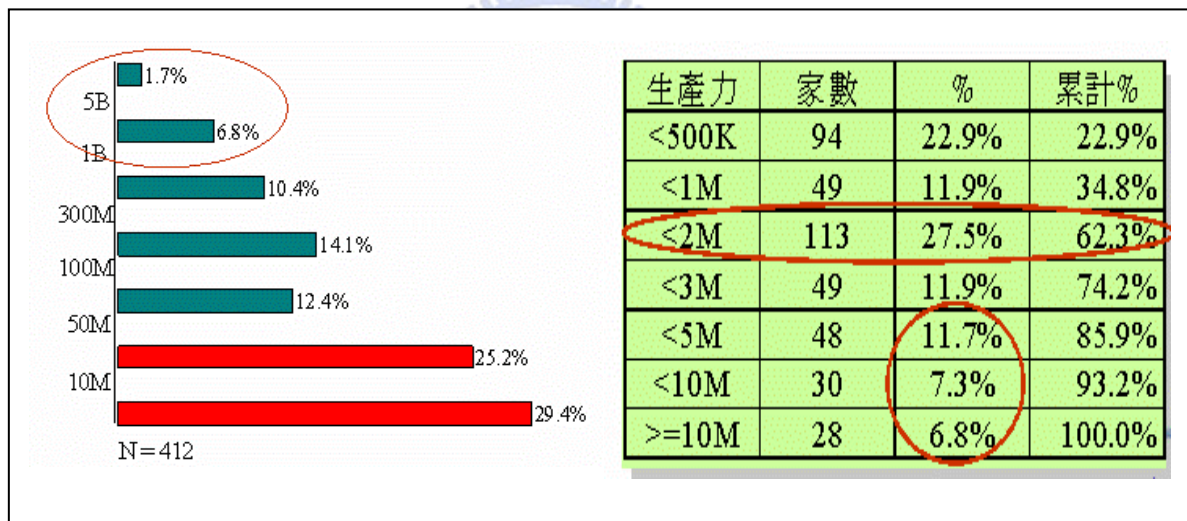


圖 3：資訊軟體廠商員工人數分佈百分比 (資策會 MIC 情報中心，2002/12)

宏碁電腦董事長施振榮 [3] 指出，台灣硬體製造業發展至今 22 年，累積相當豐碩的國際成就，反觀軟體產業，40 年下來產值還非常小，一堆軟體公司賺不到錢，這等於軟體產業沒有希望，目前軟體業的 Business model 有很大問題。台灣軟體業的問題在於沒有標準開發方式，習慣單打獨鬥，無

法承接大型專案，資訊軟體業的問題是十幾年累積下來的，依資策會情報中心 92 年研究報告 [4] 指出，台灣資訊軟體業技術能力偏低且存在下列現象：(1) 市場惡性競爭，形成技術反淘汰現象；(2) 內需市場小，缺乏市場誘因以提昇各項技術能力；(3) 無明確的產業標準引導廠商技術發展，不利於分工體系建立；(4) 人力流動快，不利於技術能力蓄積。國際競爭力不足的問題，來自於軟體產業規模不大無法像製造業一樣形成供應鏈，製造業常以完整的供應鏈面對國際競爭，競爭能力自然大增。軟體產業無法形成供應鏈的原因是軟體開發製程沒有標準化，無法有效的分工，在大型化複雜度高的專案常無法發揮綜效。[10]軟體公司成功關鍵之一為：如何了解市場需求提供所需功能，保持最佳品質，同時在最短時間內搶到占有率。對軟體公司而言，生產力與品質是軟體成功的關鍵[11]，而成本控制與時程掌握是公司經營績效的主要因素[12]。因為缺乏標準化製程，許多軟體開發便倚重人力，不同的人員素質產生不同的品質，因此品質不一，成本無法控制，時程也缺乏可預測性。除了本身軟體開發的問題，軟體業者面對大陸本地市場與印度兩大國際軟體業的競爭。此外印度軟體公司大舉進軍台灣，搶奪金融業、電信業的委外軟體服務商機。對於國內的軟體業者體質不佳又要面臨國際競爭，大陸是新興市場軟體成本相對低廉，除了大陸軟體市場的誘因外，在大陸利用較低成本建立軟體代工廠，也是許多軟體公司面臨競爭時的另一選擇。

中國大陸已經漸漸站上世界經濟舞台的中央，根據 IDC 研究指出，2003 年大陸軟體市場規模為 21.1 億美元，較 2002 年成長了 15.4%。預估 2008 年大陸軟體市場規模將達到 58.2 億美元，2003 年到 2008 年的年複合平均成長率為 22.4%。2003 年以應用軟體市場佔整體軟體市場的 48.7% 為最大市場區隔。大陸所開發的軟體成本比印度約低廉 20%，加上大陸積極發展軟體產業，頻頻向印度取經，積極培育人才，預期將於 5~10 年後擠下印度，成為僅次於美國的全球第二大軟體出口國。中國大陸將建設十個國家軟體產業基地，包括了成都、北京、上海、大連、西安、濟南、杭州、廣州、長沙、南京，並積極和大學以及大型外商軟體公司展開合作。印度一半以上的軟體公司都已通過世界最頂級的 CMMI 第五級軟體能力認證，是公認的軟體大國。印度在科技業的優勢，仍是目前大陸無法與其競爭的一環 [1]。印度擁有美國以外全球最多的英語軟體工程人才，經營成本卻只有美國的 25% (圖 4 所示)。歐美各國軟體廠商在不景氣下，為求進一步降低營運成本，已將許多軟體專案改為委外，交給印度公司負責。根據統計，已有超過 185 家 Fortune 500 大公司將專案交由印度公司代工。印度現有超過 3,000 家軟體公司從事軟體代工，其中前 25 大軟體服務公司約佔 60% 的軟體出口產值。

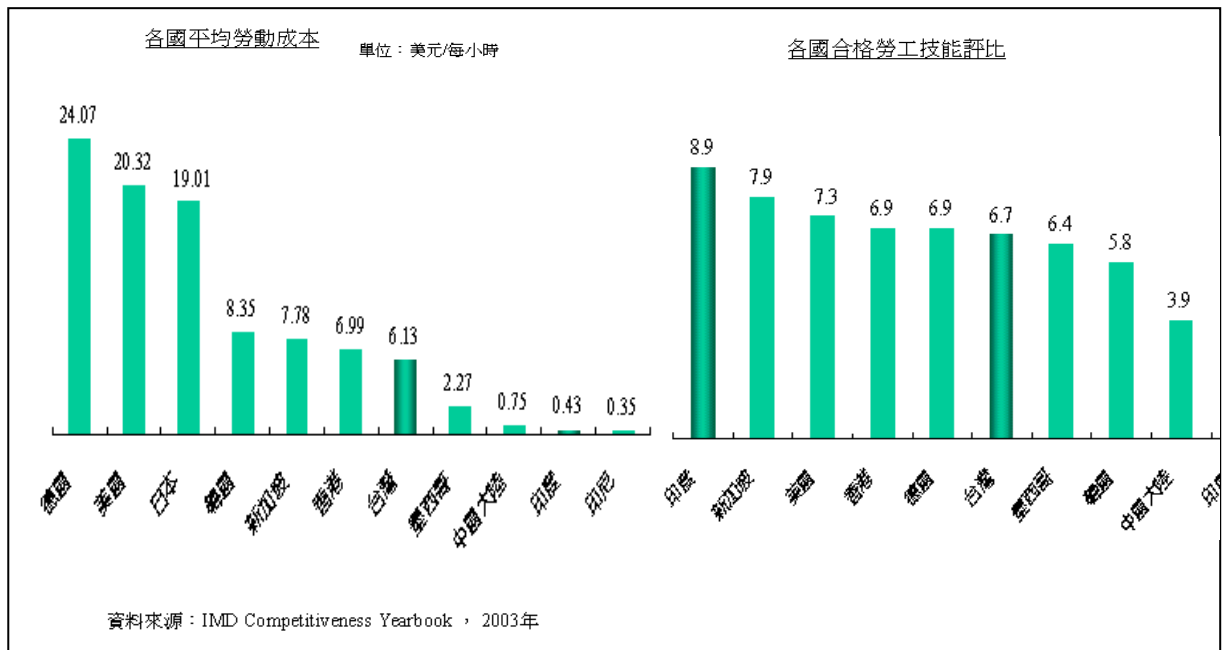


圖 4：各國平均勞動成本與技能評比

印度於過去 10 年在政府及產業界共同努力下，已建立具高度競爭力的軟體及資訊服務業，從 1995 到 2000 年的五年間，印度整體資訊產業複合成長率為 42.4%。2000 年印度整體資訊產業產值為 86.7 億美元，其中軟體產業又佔印度 IT 產業總產值的 65%。印度軟體產業整體實力在全球已佔有一席之地。軟體開發的問題是普遍性存在的，國外 Standish Group 從 1994 年起針對 30,000 個專案樣本作研究調查分析。[14]，可以看到專案超過時程與超出預算成本的問題並未改善，即使軟體工程技術已經有進步，仍然只有不到 30% 的專案是成功的（圖 5 所示）。

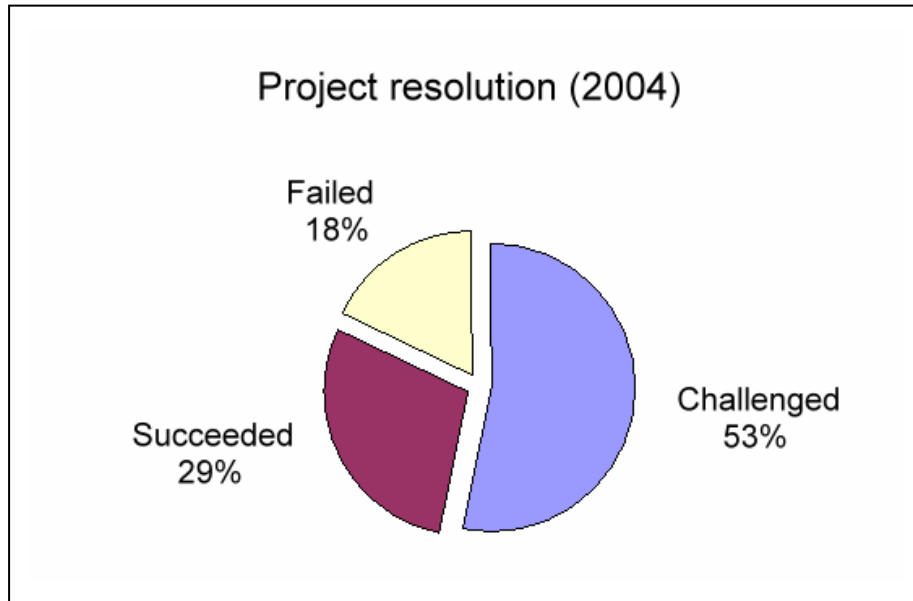


圖 5：專案調查分析 (Standish Group, 2004)

以 2004 年為例，有 29% 的專案成功，18% 專案失敗，53% 專案超過時程與成本。如果從歷年統計資料來看失敗和時程遞延、成本超支的專案總比例超過 70% (圖 6 所示)。

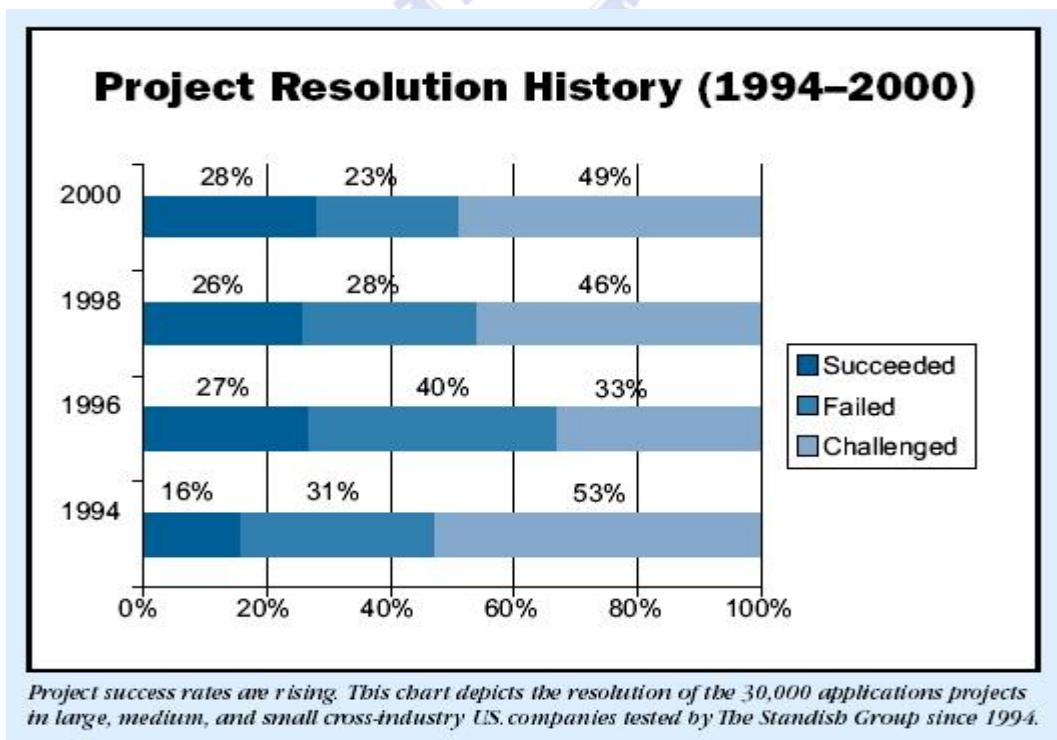


圖 6：專案調查分析 (1994-2000) 資料來源：Standish Group

根據與各軟體業相關人員訪談結果，國內軟體公司開發主要面臨問題（圖 7 所示），大約可歸納成人員素質不一致、基層開發設計流動率高（流動率約 18%），客戶需求變動多、重工比率高（50%-60%）、專案複製率低、成本估算不準確，這些問題造成的影響到“時程不確定”、“成本不確定”、“品質不確定”。

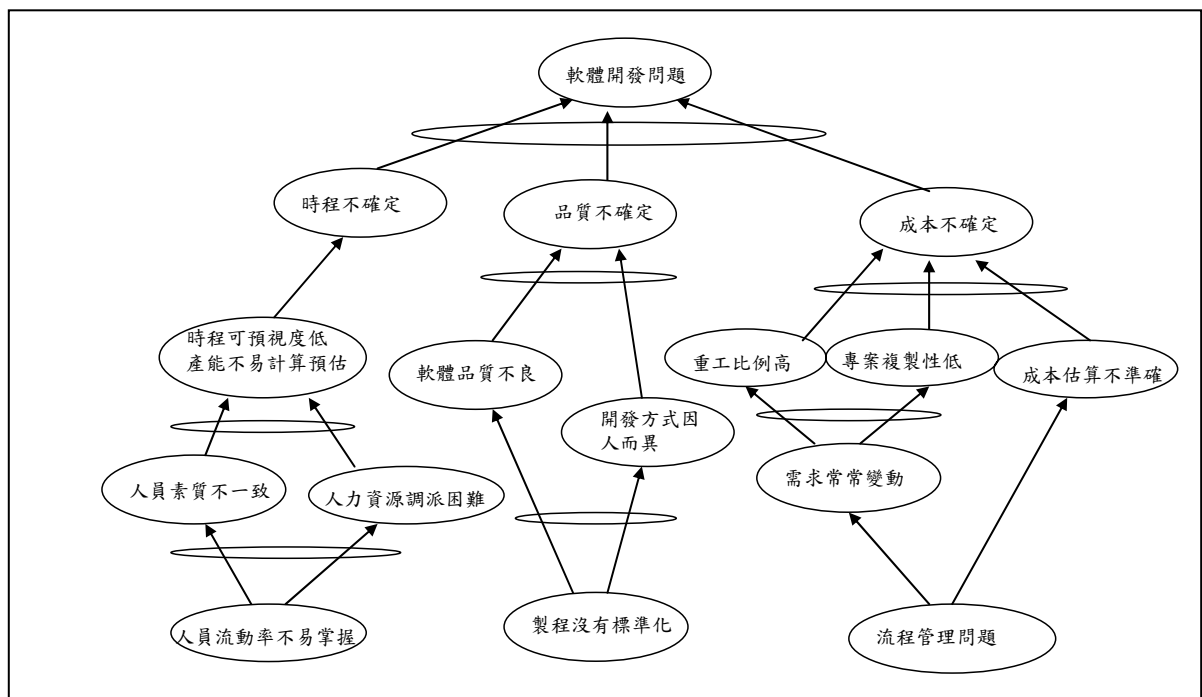


圖 7：軟體開發面臨的問題

1.2 研究目的

本研究希望能從製造業生產的角度來看軟體開發問題，以提出解決方案，硬體製造業對於製程標準化與效率分工有許多成效，譬如 IC 產業分為設計、製造、封裝、測試。而軟體業者通常是由一家公司負責需求分析、系統分析、系統設計、編碼、測試、安裝保固等全部流程。軟體一直被認為是

藝術成份多於工程，許多是“人”的問題，大多數的軟體公司都是依靠幾位明星工程師，人員的素質並不一致，產品開發時程侷限於少數人。沒有標準化製程，就無法確實掌握軟體開發時程、開發成本。也無法縮短軟體交貨時程。有標準化製程，就容易分工，多數的軟體公司只能單打獨鬥。要整合成供應鏈必須每個界面都能標準化，而且要有流程控制與自動化生產的工具。

1.3 研究方法

本研究所採用之研究方法，有文獻探討與個案研究。文獻探討主要就目前相關軟體工廠、軟體元件等文獻，作一分析探討。找出可以參考的軟體工廠解決方案。參考案例則以國內上市的系統整合電腦公司所屬的軟體工廠為研究對象，個案公司分別在大陸、台灣都建置軟體工廠，承接軟體代工設計與編碼，是一個標準分散式軟體生產模式。透過個案公司來了解目前建置軟體工廠的做法與方向。

二. 文獻探討

2.1 軟體工廠

軟體工廠 (Software Factory) 是軟體工程 (Software Engineering) 中的一個研究方向，依不同地區特性與經驗大約可分為幾個不同類型 [16]，日本的軟體工廠研究以電腦與通訊產業的製造商為主，如 NEC, Toshiba, NTT 等公司，以強調製造生產的組織與紀律 [23, 24]。歐洲 Eureka 軟體工廠研究專案 [32, 33, 34]，是一個從 1987 年到 1996 年，共 2400 人年的研究計劃，成員包括電腦製造商、軟體公司、大學等單位，計劃目標為是以提供技術、標準、組織為基礎，建構一個軟體工廠共用架構，參與計劃成員以共用架構開發元件與軟體，讓不同的應用軟體可以透過軟體通道互通。美國軟體工廠研究最初是以經驗庫知識 [35, 36] 及標準架構 [42] 及軟體成熟度 (CMMI) 為主。中國大陸以小型生產單位導入 ISO 9001 與 CMM 做為研究 [31]。軟體工廠主要是將其他產業的管理概念移植到軟體開發，以工廠嚴謹的生產管理方式，來改善軟體產業的生產力與品質。製造業追求品質持續改善的成效如果可以移植在軟體生產中可以解決多年來的軟體問題。最早的軟體工廠概念是 1968 年美國奇異 (General Electric) 公司的工程師 R.W. Bemer 建議奇異公司設立軟體工廠 [15]，目的是降低程式設計師之間的生產力差異，使用的方法是採用標準化工具，電腦化介面管理暨儲存財務與管理的歷史資料庫。雖然奇異公

司於 1970 年底退出商用軟體與硬體的市場。Bemer 先提出了軟體工廠第一個重要的概念：標準工具 (Standardized tools) 與控制 (Control)。用標準工具來來控管產能與品質，用電腦化財務與管理資料來追蹤成本和時程。

1980 年代日本的電腦與通訊廠商紛紛加入軟體工廠的研究，其研究方向是將生產製造的概念製轉換到軟體開發，主要軟體工廠的重點有：(1) 建構一個支援軟體生產的流程；(2) 建構軟體作業平台；(3) 建構組織來監控軟體生產流程；(4) 建構標準化工具、介面、維護、標準設計流程；(5) 型態管理 (Configuration Management)；(6) 教育訓練；(7) 專案管理系統；(8) 成本管理系統；(9) 產能管理系統；(10) 品質保證系統；(11) 品質衡量機構；(12) 品質循環活動。1980 年日本 NEC 公司建置軟體工廠 [17]，為了要改善軟體生產力和品質，NEC 的軟體工廠包含了：(1) 軟體開發生產線 (Production Line)；(2) 管理系統；(3) 資訊系統；(4) 電腦與通訊開發環境；(5) 人力資源；(6) 軟體品管小組 (圖 8 所示)。

1988 年 Nunamaker 提出提昇軟體生產力的四個方向 [30] 為：(1) 自動化軟體開發工具；(2) 考率軟體開發過程中人的因素；(3) 評估與改善的資訊系統；(4) 軟體工廠。1991 年 Humphrey [19] 認為軟體工廠要有標準生產工具、方法與再使用元件 (reuse software component)，同時工廠的紀律不容易應用在軟體開發上。

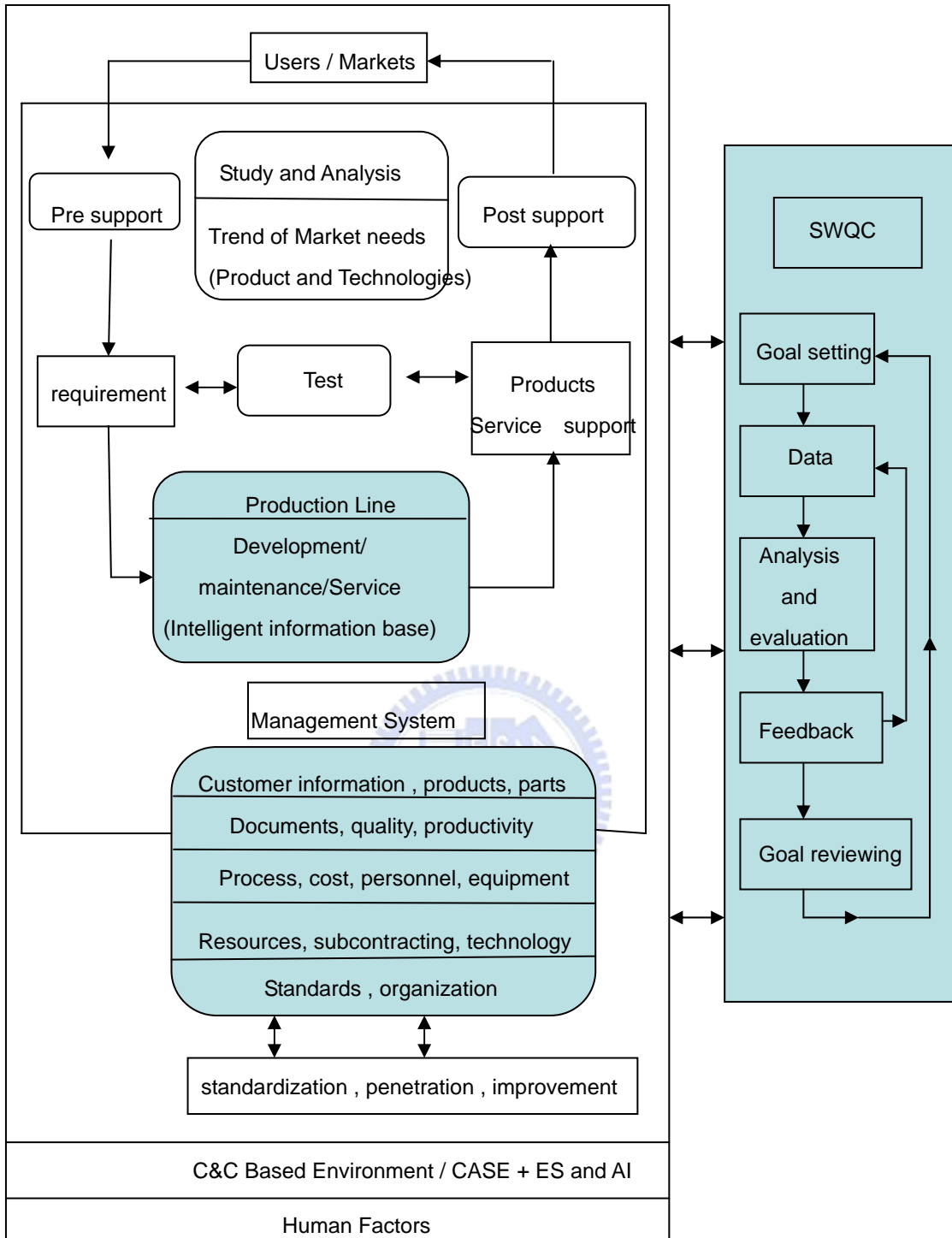


圖 8 : NEC Advanced software Factory (資料來源 : NEC corporation)

2.2 軟體元件再使用

1968 年 AT&T 貝爾電話實驗室 (Bell Telephone Lab.)，Dr. M. D. McIlroy 在 NATO Science Committee 提出另一個概念 “在開發新程式時使用舊程式碼” 以達成大量生產 (mass production) 的目標 [18]。McIlroy 啟發了軟體元件再使用與元件工廠的大量生產概念。1977 年日本 Toshiba 公司為了開發工業流程控制系統軟體 [26, 27]，成立 FUCHU 軟體工廠 (圖 9 所示)，Fuchu 軟體工廠的組織是以高度軟體再使用為目標，部門分成元件製造與專案開發，在兩部門之上成立軟體元件指導委員會 (Software Part Steering committee, SPST)。

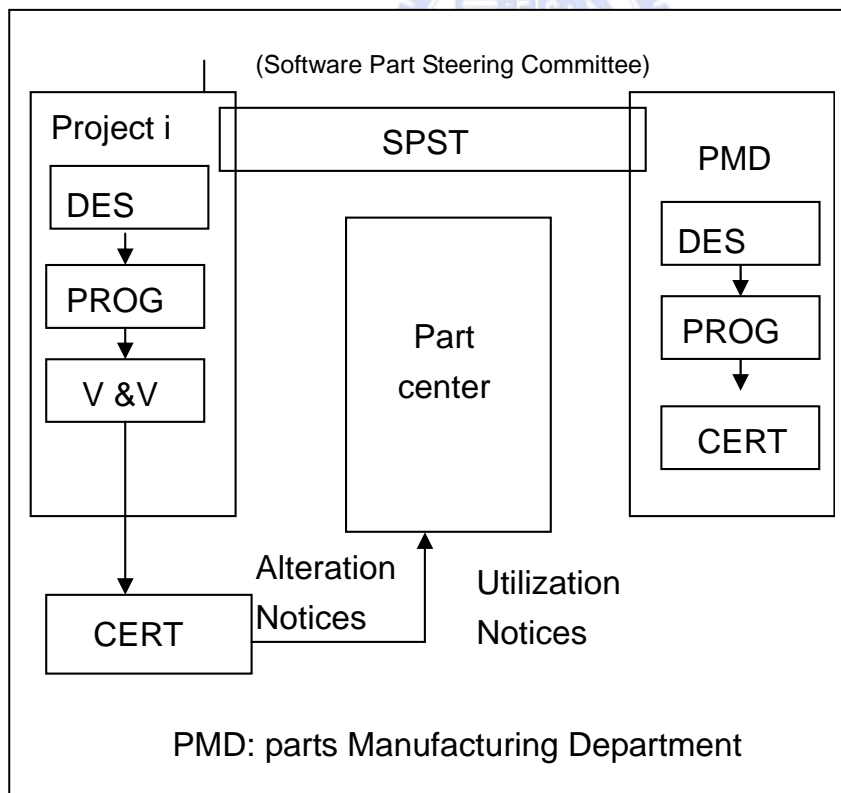


圖 9：Toshiba Fuchu SF

元件製造部門負責軟體元件設計、生產、測試，元件中心為一個軟體

元件資料庫，提供專案開發所需的軟體元件。Basili [35, 36] 提出重複使用開發經驗的方法，以建置經驗庫來提昇軟體品質。專案開發過程都會參考經驗庫（圖 10 所示），並以經驗庫中的完設計文件或軟體為下次開發基礎。經驗庫資訊必需分類、權限管理，方便儲存、搜尋、擷取。

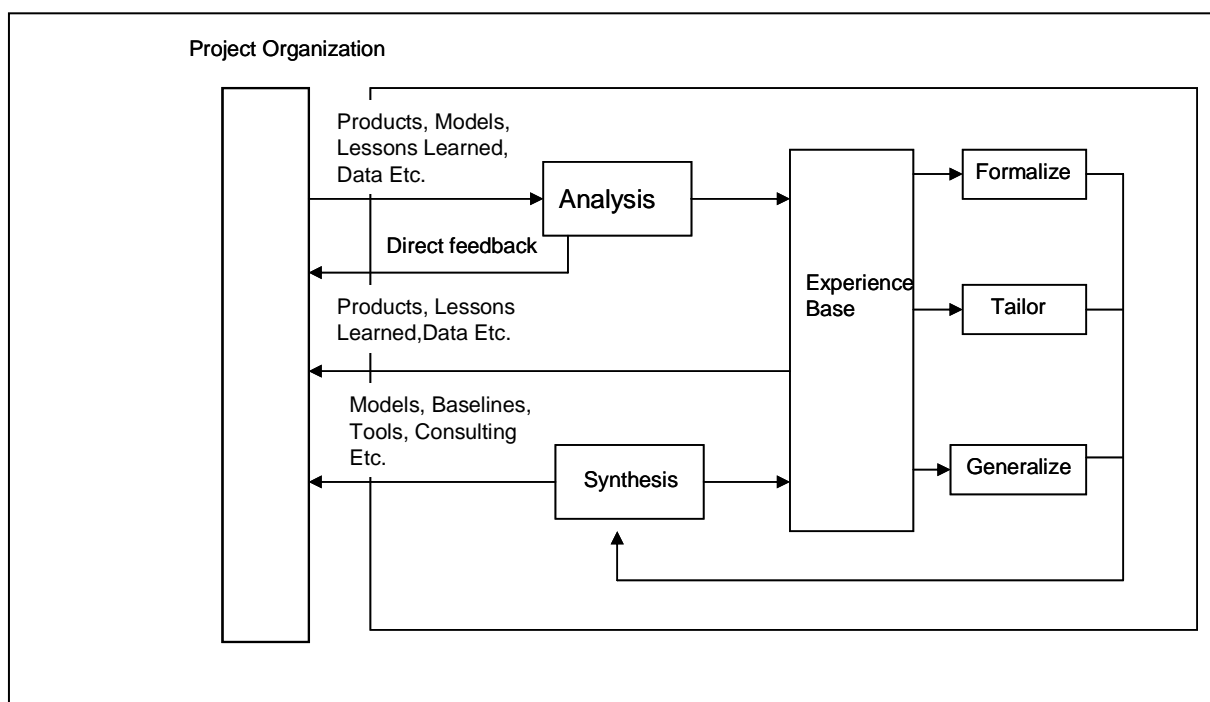


圖 10：經驗庫軟體開發模式

1992 年 HP 軟體實驗室提出以軟體元件(圖 11 所示)組合概念的彈性軟體工廠(Flexible software factory)[37]，以不同產業別的軟體元件組裝應用軟體，軟體開發前先做產業分析、系統設計、產品線生產等步驟。(圖 12 所示)，完全類似工廠生產模式。

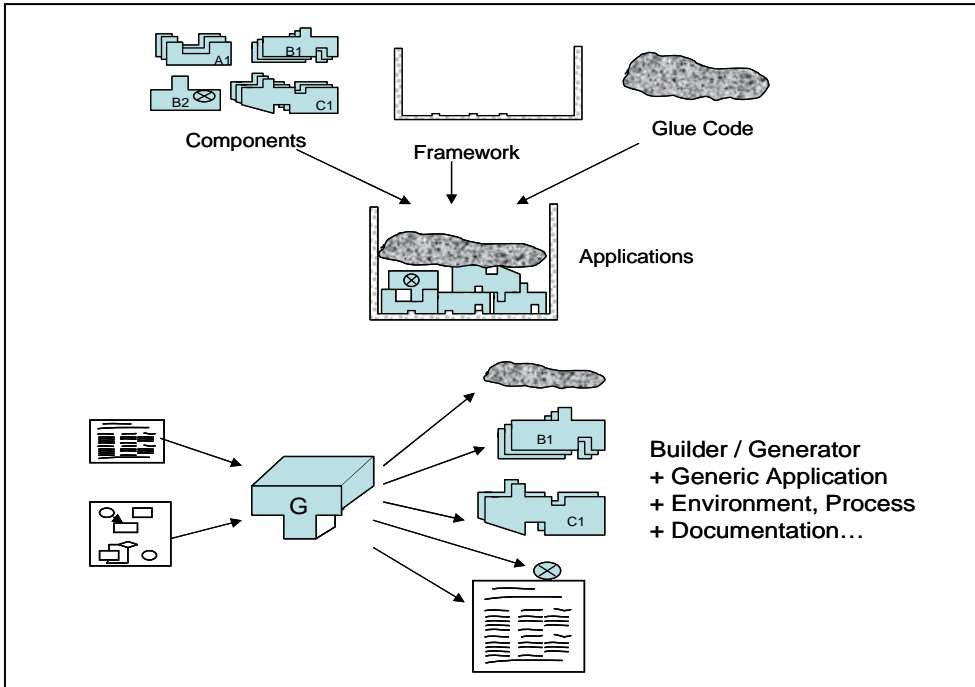


圖 11. HP 軟體實驗室 - 產業別應用元件

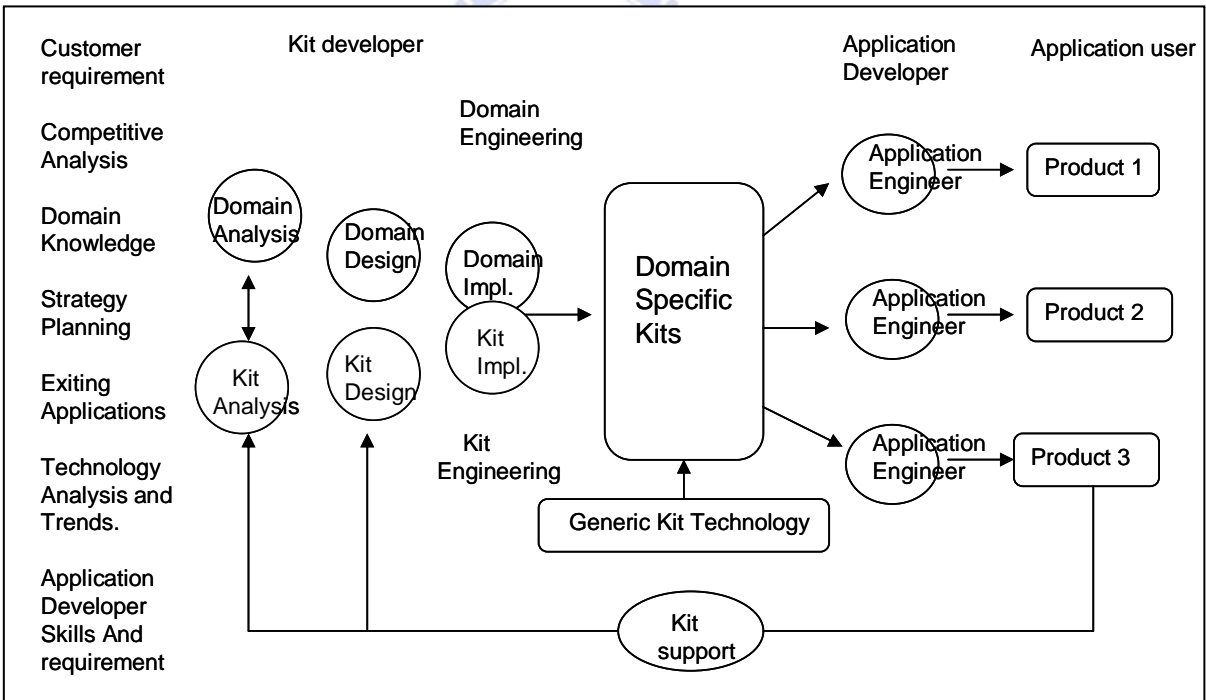


圖 12. 元件組合軟體開發

三. 軟體工廠生產模式

3.1 軟體工廠解決策略

軟體開發流程依序分為需求分析、系統分析、系統設計、程式編碼，測試、維護。其中可以大量自動化的部份從系統設計到程式編碼、測試、維護。[21][22][25]這部份是軟體工廠主要涵蓋的部份。在建置軟體工廠前需考慮到軟體生產工廠和製造工廠的差異性（表 1 所示），製造工廠管理的重點對像是“物料”，而軟體工廠管理的對像是“人”，以生產製造的角度來看軟體工廠的管理，著重的是“標準化”和“作業程序”。軟體工廠是將製造業的生產與品管概念，應用到軟體開發流程，以達到人員生產力，時程、品質的一致性與可預測性。其所需要規劃建置的工作有標準化的軟體製程與生產中心，軟體元件庫，自動化工具支援等。利用軟體工廠生產模式可以做到下列三項目標：(1) 降低人的因素，品質一致；(2) 軟體模具與元件再使用 (re-use)，開發時程短，專案成本低；(3) 製程標準化，可以分工，容易計算時程與成本 [28][29]。在國外大型專案軟體開發也有許多應用工廠製造概念[40][41]。

表 1：軟體工廠和製造工廠的差異

	軟體工廠	製造工廠
產品	無形與有形(文件)	有形
產品儲存性	低	高

產品一致性	低	高
對勞力依賴程度	高	低
投入 (Input)	人力	物料
轉換程序 (Process)	腦力密集，不易標準化	資本密集，容易標準化
產出 (Output)	程式碼	實體
回饋 (Feed back)	QC 不易進行	QC 容易進行

因此軟體工廠的重點應包含：軟體生產製程、元件庫、自動化生產工具、管理工具支援環境。



3.2 軟體工廠組成

依照軟體工廠概念軟體工廠應該有下列組成（圖 13 所示）。設計研發單位負責生產線設計、元件製作、軟體模版設計；生產管理單位負責生產規劃、派工、排程、外包管理；生產中心則負責元件組裝，編碼；品管單位負責單元測試、整合測試。從功能模組方塊來看（圖 14 所示），軟體工廠投入的有程式規格、資料模型規格、使用界面規格，透過生產中心依照模版與元件組成合於規格的程式碼，並作完單元測試與整合測試，產出程式碼與文件，產出屬於非實體的數位格式，需存放在儲存媒體上。由於產

出是數位格式資料，方便於網路上直接傳輸，且容易被修改造成版本混亂，因此需要型態管理（Configuration Management）來控制規格異動與版本異動。

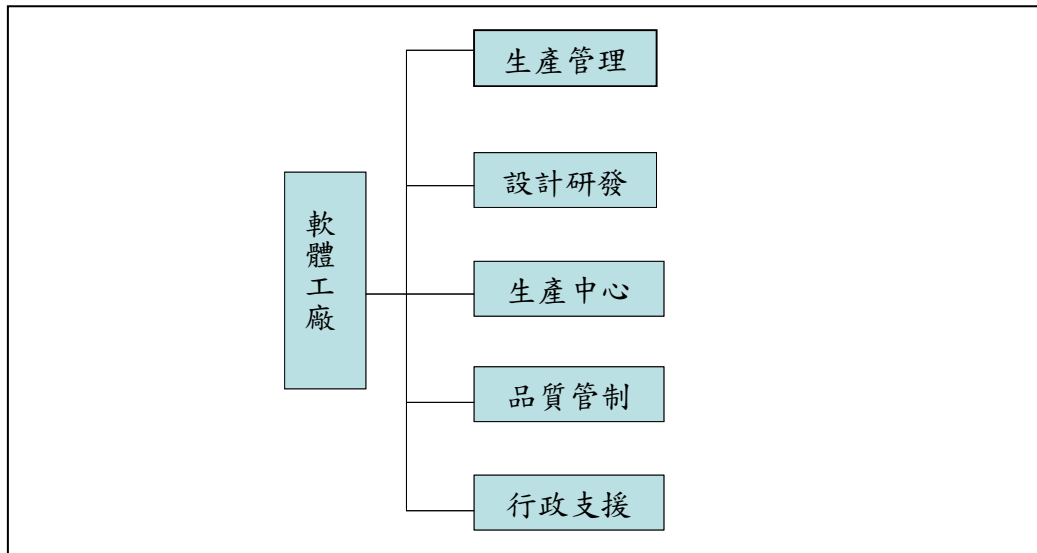


圖 13：軟體工廠組成

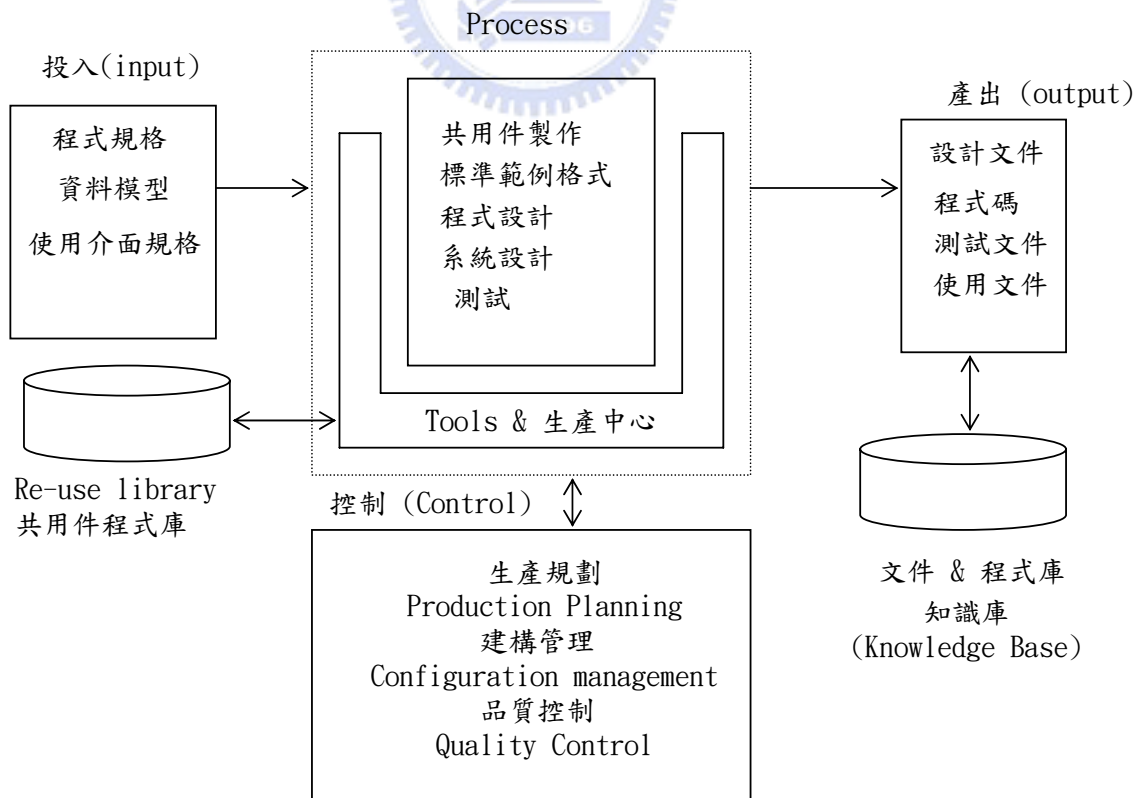


圖 14：軟體開發各階段產出

3.3 從訂單到派工的製程規劃

軟體生產中心主要為元件組裝與編碼，為了規劃產能與計算成本，軟體工廠需依程式等級標準來分類（表 2 所示），作為生產中心成本與績效管理的標準

表 2：程式等級與標準工時

程式等級	標準工時(人天)	功能點數	備註
A 級	5	20	核心交易
B 級	3	10	一般交易、流程
C 級	1.5	7	報表
D 級	1	5	建檔(新增、查詢、刪除)
E 級	0.5	3	功能修改
F 級	0.25	1	功能修改

從訂單承接開始必需先完成生產規劃，將訂單展開成工作分解結構 (Work Breakdown Structure)。從訂單分解成工作 (Task)，次工作 (Subtask)，從次工作分解成基本工作單元 (Work Unit)。每一個工作單元即是單一工作者可以獨立完成的部份。依據工作單元與產能狀況，決定生產排程。生產規劃內容包括：程式數量、程式規格、完工日、使用工具、

程式模版、測試標準、效能標準，生產計劃確認後產生外包計劃或派工單。

軟體工廠所需之流程控制與管理模式，從訂單計劃到派工管理、元件規劃、模版製作、編碼、測試、重工與外包的整個製程。這些過程都需要控制及管理，且需借助於下列功能。

1. 生產規劃：生產依訂單型態分成兩種：按訂單生產及元件生產，提供最程式等級程式規格、完工日、使用工具、程式模版、測試標準、效能標準。
2. 生產排程：用有限或無限產能，進行各工作站進度排程，視需要再前後調整，另外也進行訂單模擬，以訂單交期反推至各工作站，該站產能比對，若超出產能限制，視狀況發出外包申請。
3. 製程工單：列出生產工作途程，及優先次序，元件及測試標準、效能標準。
4. 品質管制：可即時查詢訂單製程狀況，並收集在現場上的生產數據，包括重工及不良品數量與原因，元件使用率，生產效率及作業成本。各部門之相關職掌如圖 15 所示。

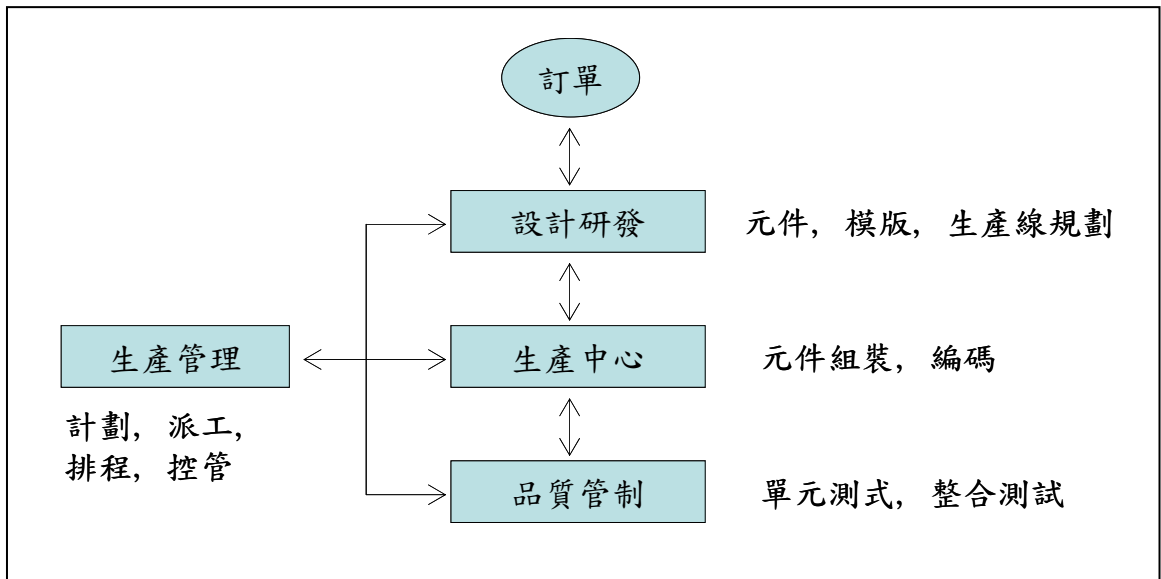


圖 15：軟體工廠部門職掌

3.4 生產中心的規劃

軟體生產中心的使用自動化工具，軟體模版概念可以大量生產，降低學習技術門檻，使用元件庫可以降低成本，從製造業生產的經驗研究發現：標準化製程、模組化元件、自動化工具、品質控制，是生產管理中重要的關鍵，軟體生產過程中的產出都是可以數位化的資料，可以完全沒有實體，一直到最後交付階段才使用光碟當載具把軟體燒錄在裡面，做為交付過成的實體。軟體工廠上有不同的工作內容規格：(1) 程式規格；(2) 資料模型；(3) 使用介面規格，這些規格為生產中心的投入（圖 16 所示）。一般上游需求分析與系統分析完的產出有兩大類方法論：結構化分析與設計與物件導向分析與設計。

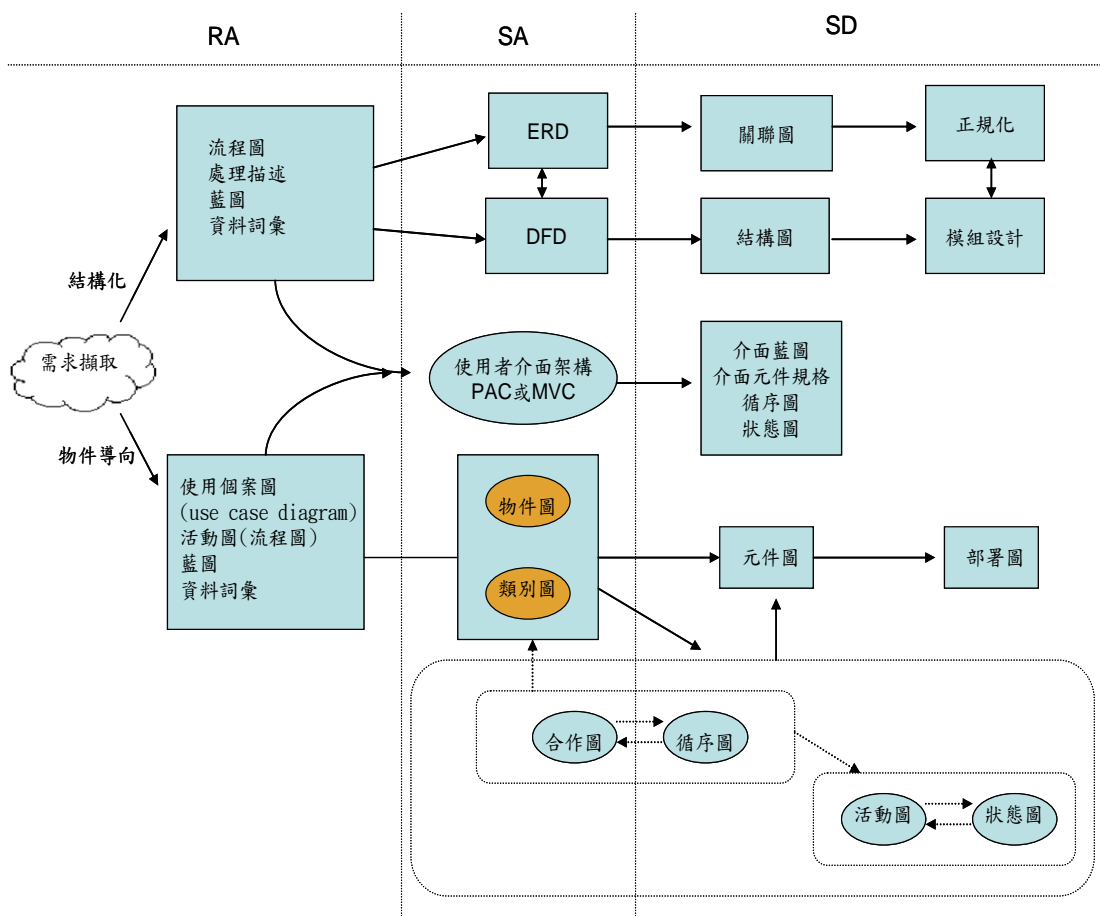


圖 16：軟體工廠上游規格

資料來源：[13]

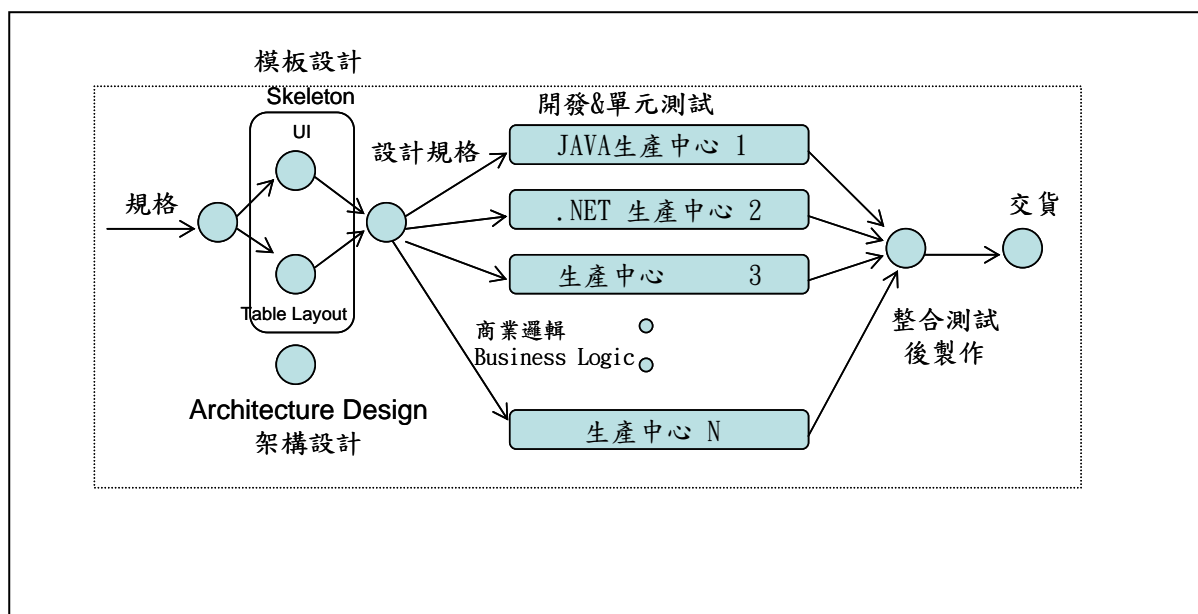


圖 17：生產中心規劃

這兩類方法論的設計文件產出完全不同，但是都可歸類為 程式規格、介面、資料模型三項。生產中心的配置原則上以開發工具特性或產品特性為分類標準，依不同狀態可以做彈性配置（圖 17 所示）。傳統專案開發，每一次都是從新編碼不考慮重複性，通常使用複製舊程式再依差異做修改。軟體工廠重要觀念是重複使用元件、設計、架構，再使用軟體元件，軟體元件可依其層次、大小可分類（圖 18 所示）。設計樣版（Template）為不同產業的系統模組，框架（Framework）是系統設計的架構，如 CORBA、J2EE、.NET 都是標準框架，骨架（Skeleton）描述介面與基本元件規格。每一個可重複使用的軟體必須可管理，以資料庫方式儲存（Store）、擷取（Retrieve）。

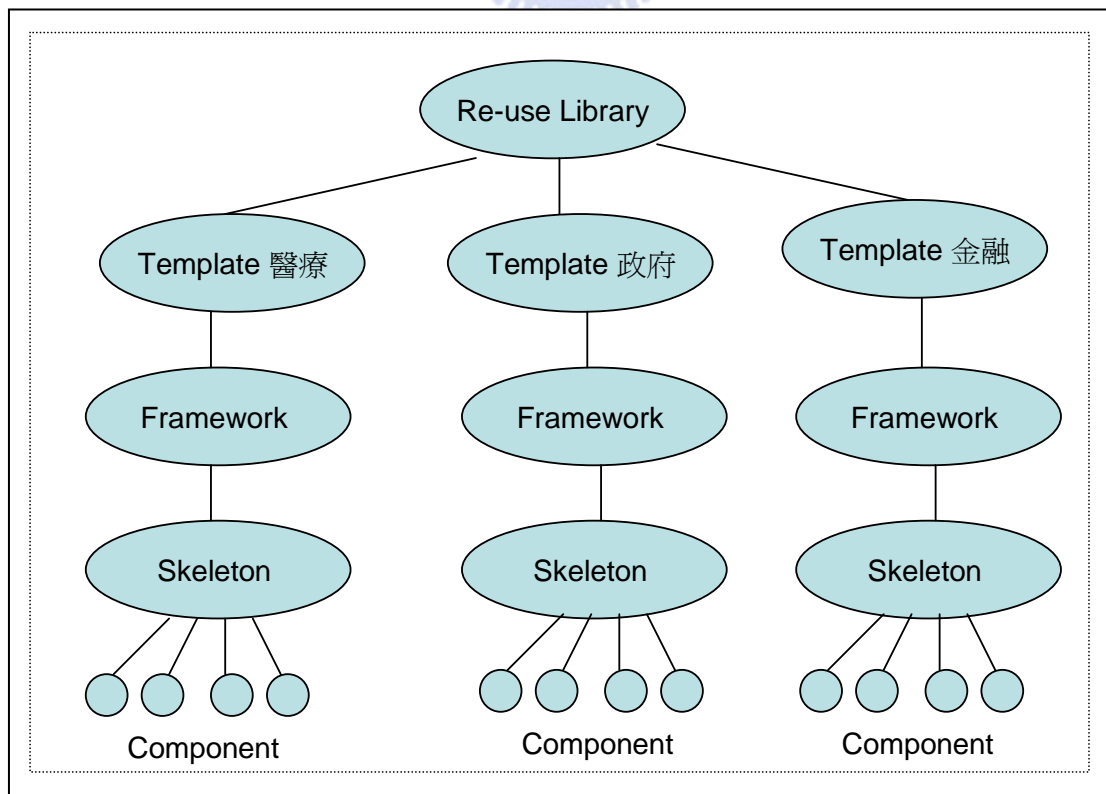


圖 18：軟體再使用元件階層

3.5 軟體模版設計與標準化工具

軟體工廠的兩項主要工作是架構設計與程式碼生產，軟體模版是生產工具用來產生程式樣式的基本雛形定義。為了讓設計與編碼工作可以大量自動化必須依靠工具，軟體生產中心使用生產工具來達到標準化與自動化的目標。軟體工廠需要設計工具、測試工具、專案管理系統 (Project Management System)、品質管理系統 (Quality Management System)、型態管理系統 (Configuration Management System)，系統工具必需能整合在同一個使用者介面 (圖 19 所示)。

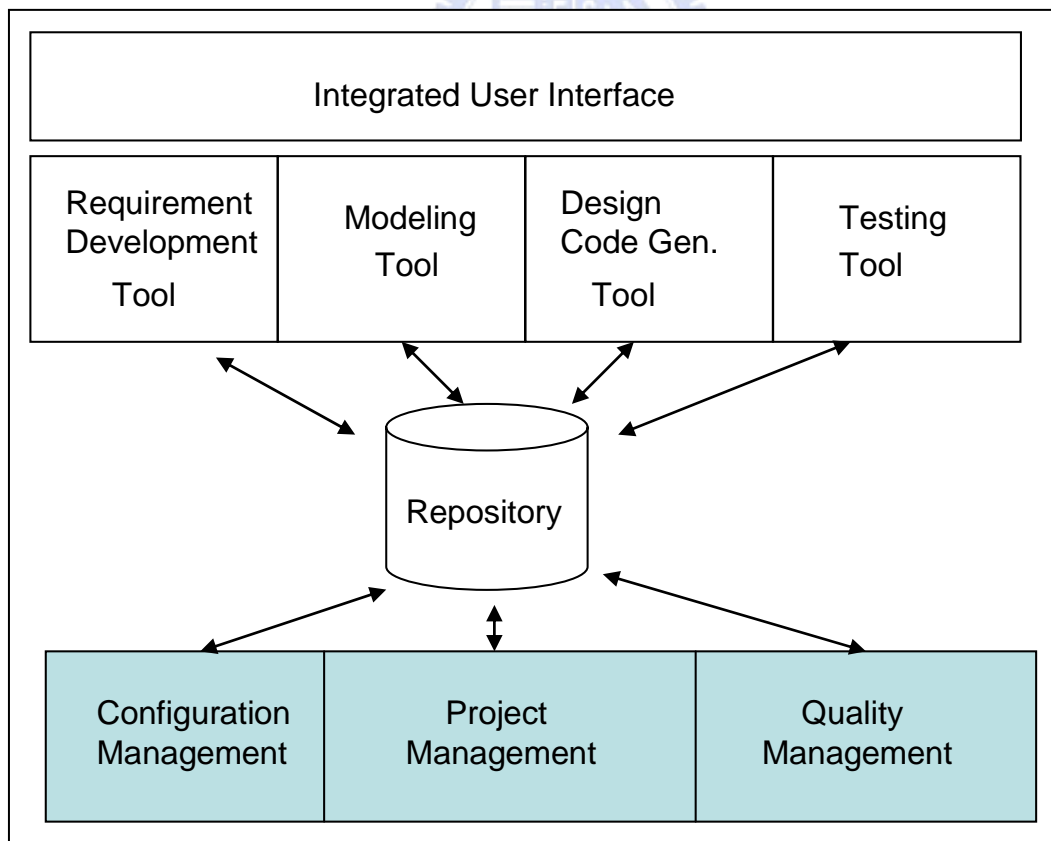


圖 19：生產工具整合

四. 軟體工廠個案

4.1 個案公司背景

個案公司成立於 1975 年，主要從事系統整合與軟體專案服務，初期以代理國內外週邊設備為主，1979 年起開始針對證券、銀行、醫院、政府、電信、軍方等單位提供系統與軟體建置服務。於 2001 年 5 月於台灣證券交易所上市，2004 年 6 月 25 日通過 CMMI（軟體能力成熟度模式）第三級評鑑，定位企業核心競爭力於軟體與服務。系統整合與軟體專案服務是個案公司主要的核心業務，為了提高國際競爭力、增加軟體產值，在考量全球軟體產業的未來有逐漸朝向「國際化」、「大型化」發展的趨勢，目前除了在台灣地區設有六個營業據點，並將觸角伸展至海外，在全球各地皆有佈局，如美國、泰國、日本、中國大陸的北京、上海、深圳、西安、蘇州等地。其中個案公司分別於台灣和大陸西安建立了兩個軟體工廠：A 廠、B 廠。軟體工廠以軟體設計，編碼，測試的後段工作為主，承接台灣、日本等地訂單的軟體外包工作，建立起一個軟體生產供應鏈。在建立整個分散式軟體工廠生產模式之前，個案公司進行了兩項前期計劃：(1) 組織與流程改造；(2) 分散協同作業管理工具開發。

4.2 分散式軟體工廠組織流程改造

一般軟體公司組織都是以產業別來分組 [5]，個案公司原來的軟體開發部門是依照產業別分工設計（圖 20 所示），每一個不同產業的小組有自

己的工程人員：系統分析師 (SA)、系統設計師 (SD)、程式設計師 (Programmer)。依產業別分組的組織其優點是：產業知識集中，產品方向明確，人員內聚力強，便於溝通。缺點是：不同產業小組人員支援困難，人員不能共用，所需人力較多。產業分析與系統分析需要倚重產業知識，系統設計與程式設計比較倚重軟體開發技術，為了因應新的軟體開發程序將組織以功能面再造 (圖 21 所示)，主要在將同一性質的工作集中於一個部門。組織再造後 (圖 22 所示) 會影響專案人員之間的溝通，專案人員從以前的同一部門便成不同部門，行政部門如同一個人力資源庫，視不同專案需求從人力資源庫中組成專案團隊，專案團隊是一個高度彈性的組織，因此也需要依賴更有效率的管理工具與新的作業模式。

其次，為了達到大量生產與製程標準化目標，軟體設計與軟體開發部門改造為軟體工廠 (圖 23 所示)，並分割成為新公司，因為開發與設計一旦獨立分出成為另一家公司，所有的作業方式必需標準化，成本的核算考量嚴謹，符合公司經營最大利益目標。新的協同作業方式如圖 22 所示。

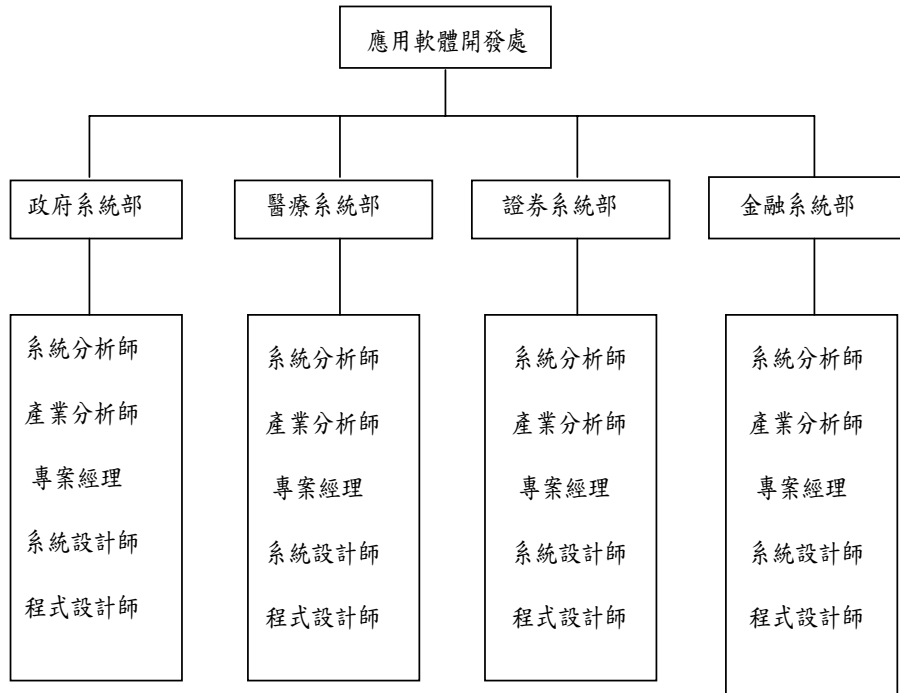


圖 20：依產業別分工組織（資料來源：個案公司）

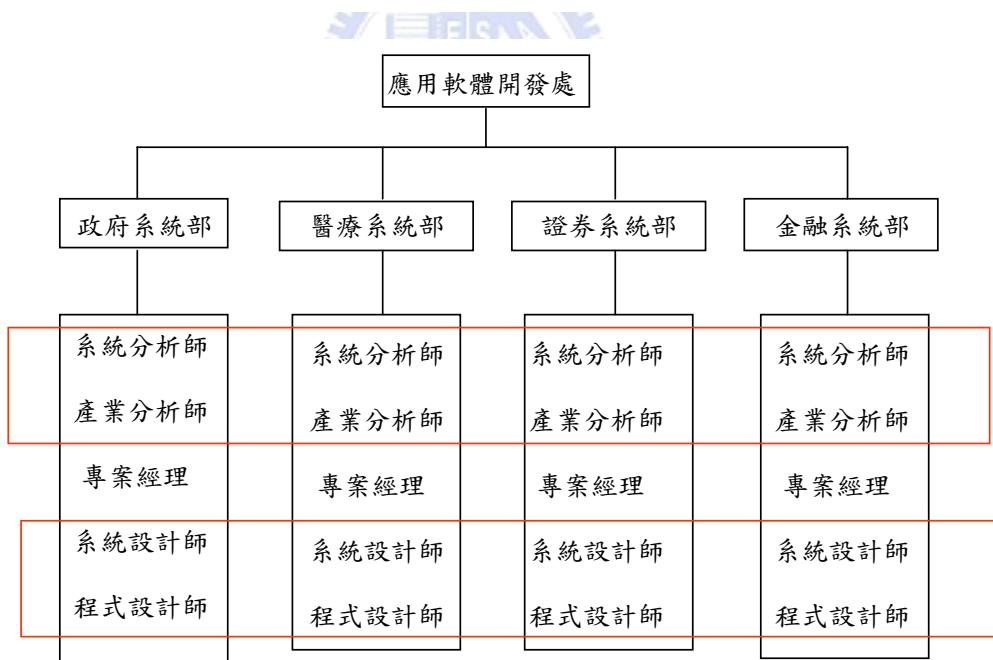


圖 21：組織再造-依功能別分組（資料來源：個案公司）

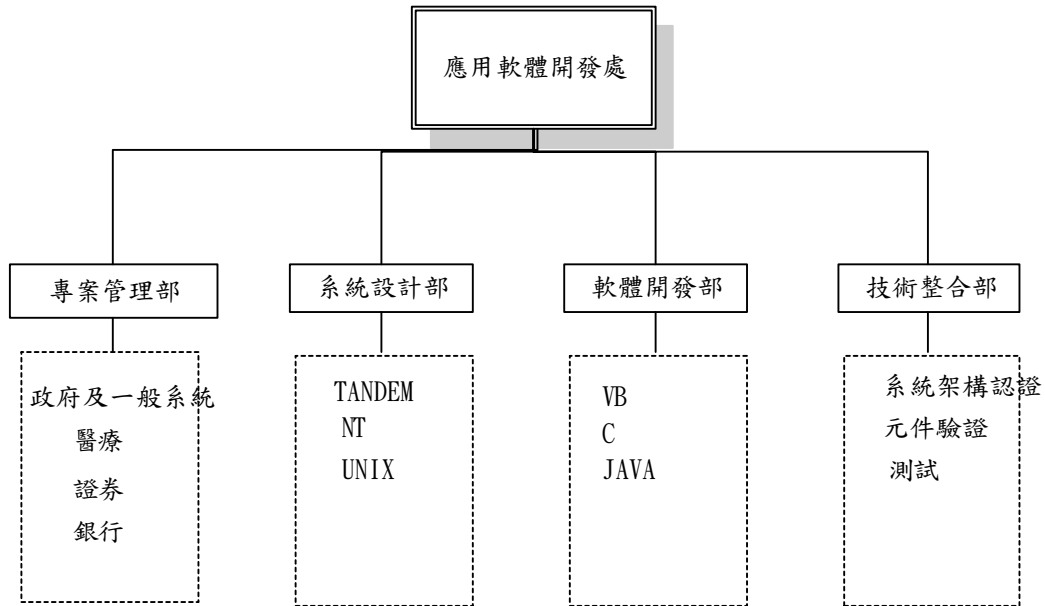


圖 22：功能別組織（資料來源：個案公司）

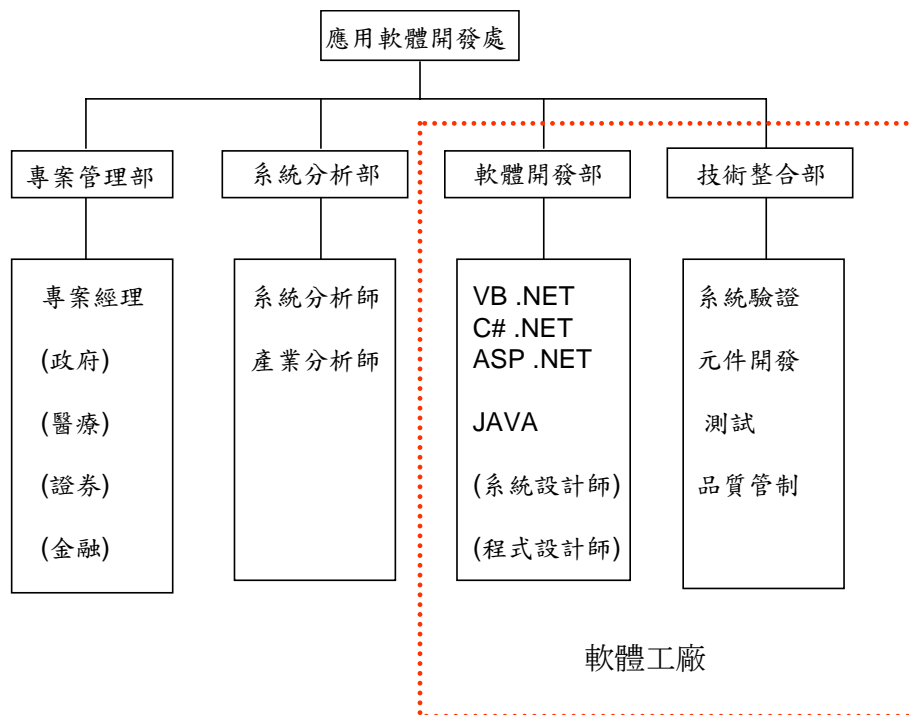


圖 23：軟體工廠組織(資料來源：個案公司)

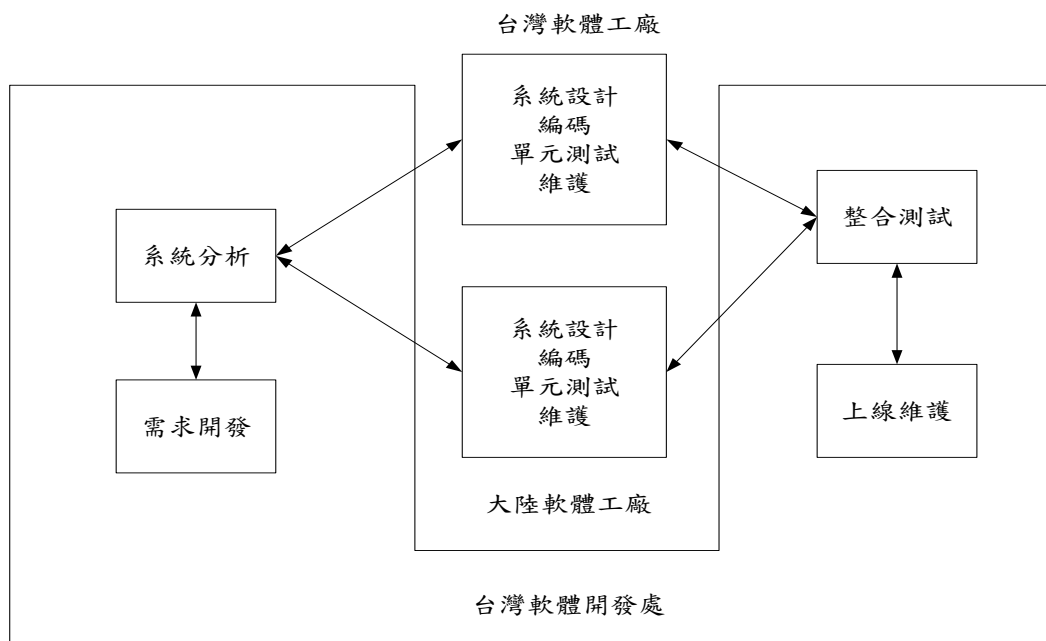


圖 24：協同作業圖

軟體工廠成為獨立公司後，增加國際競爭力，在不同地點建立軟體開發中心，並依照軟體開發模式垂直分工，形成一個供應鏈（圖 25 所示）。

同時，黃繼弘[6]認為軟體元件生產方式有助於達成國內軟體產業供應鏈管理的推行。

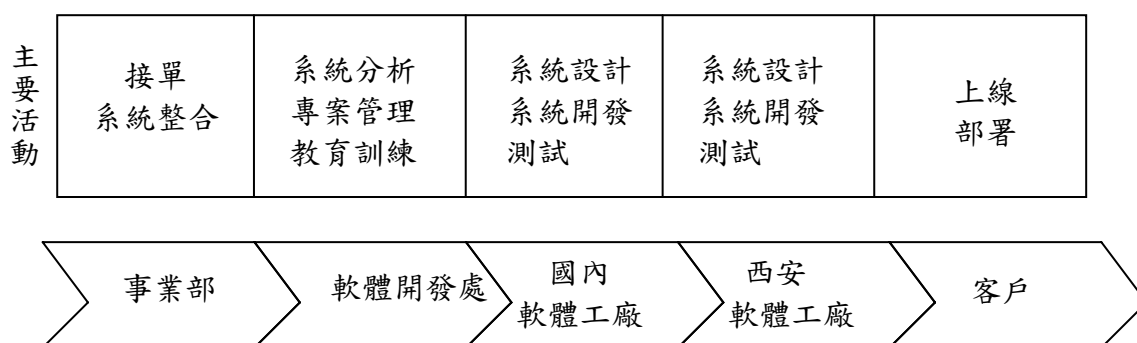


圖 25：軟體供應鏈

個案公司事業部門負責承接專案，將其中的軟硬體分別下單給不同供應商，軟體開發處承接軟體專案，並負責專案管理與系統分析工作。A 軟體工廠（台灣）與 B 軟體工廠（大陸西安）擔任系統設計與系統開發。軟體工廠為獨立運作的子公司，除了成本考量和地區考量外尚有管理模式不同的問題，成為子公司需要自行負責經營責任，可以更有效率。

4.3 協同作業平台的工具支援

軟體服務業的主要資源是“人”，因此軟體服務 ERP 被管理的對象是“人”與“服務”，有別於製造業以物料為主的 MRP。協同作業平台 Syscom Online(圖 26 所示) 是一個線上即時系統，支持個案公司電子化的運籌作業，與一般資訊管理系統的差別在於區域性、即時性 (Real Time) 及整合性。協同作業平台整合中心各點與各部門產生資料的工作流程，統一內部資料處理程序，建立企業共用資料庫並能即時使用。同時，張清峰[7]認為全球化的協同軟體研發專案管理的實施，需要複雜的作業流程，因管理流程涉及研發團隊組織結構、組織文化背景、技術整合、技術平台及資訊系統等之建構。

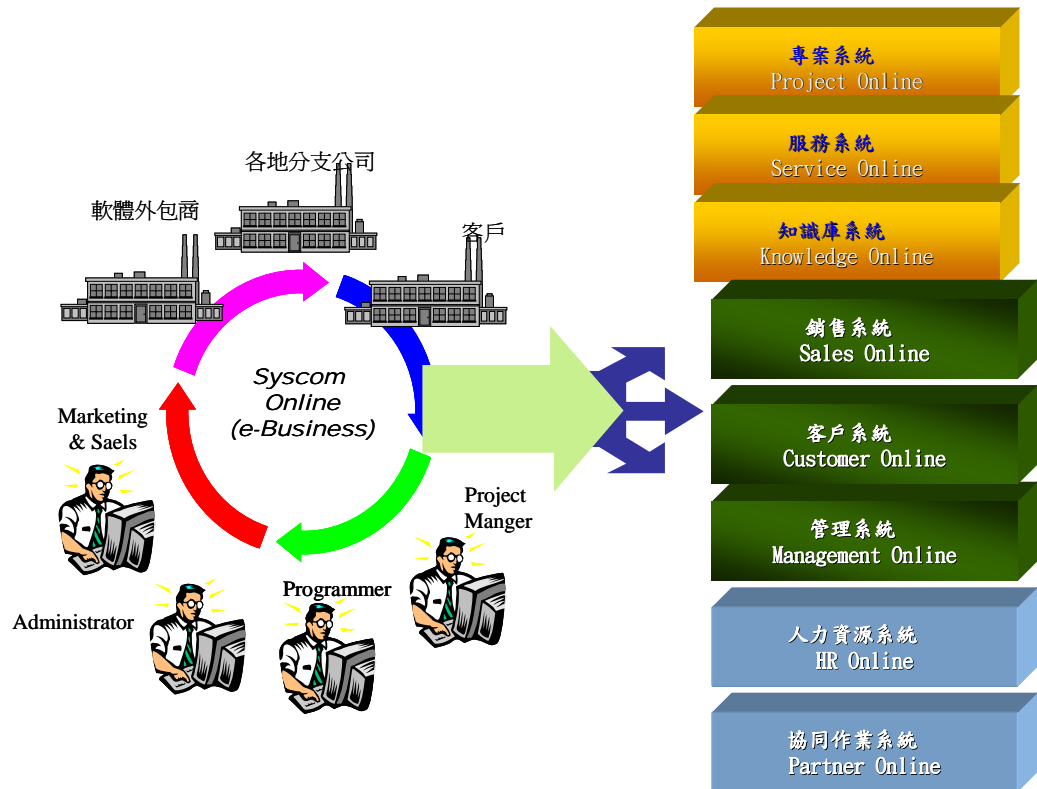


圖 26：協同作業平台架構圖

個案公司的協同作業平台子系統有：即時專案管理系統 (Project Online)、客戶服務管理系統 (Service Online)、知識管理系統 (Knowledge Online)、即時業務支援系統 (Sales Online)、客戶服務系統 (Customer Online)、即時決策支援系統 (Management Online)、人力資源管理 (HR Online)、協同作業系統 (Partner Online)，其主要特性如下：

1. 即時專案管理系統 (Project Online)

透過網路提供專案開發時的共同工作與管理平台，架構在自動化作流程引擎 (Smart Routing)，權限管理系統 (My Authority)，知識庫管理系統 (Knowledge Online) 及動態人力資源系統 (HR Online) 之上，不但即

時控管專案開發過程的資訊流，並即時掌握專案開發各階段的產出（數位化物流）透過知識庫達到智財權的管理，極大化知識與軟體的分享與再利用（Reuse）效益。

2. 客戶服務管理系統 (Service Online)

透過網路即時掌握各地的客戶服務活動，隨時追蹤是否在計劃內或合約時效內完成問題處理，並警示及 Escalate 逾時反應或處理之問題。彙集問題處理相關之經驗，建立相關知識庫，以提高問題處理績效與品質。即時客服管理系統架構在資料庫與搜尋引擎、Smart Routing、My Authority、Marketing Online、HR Online 之上提供下列主要功能：(1) 安裝計劃與管理，根據客戶採構合約並配合專案的實施進行安裝計劃及任務的分派，隨時掌握安裝之進度；(2) 維護合約管理與定期保養，建立維護合約相關資料，並據以訂定年度保養計劃及派工；(3) 客戶問題處理，建立客戶問題單，提供客戶經驗庫搜尋功能，以幫助問題的判斷與處理，並依據買賣或維護合約的相關條款隨時追蹤問題處理的時效；(4) 即時零件及備品庫存管理；(5) 客服相關統計分析，提供客服反應時間、滿意度、維修成本分析。

3. 知識庫管理系統 (Knowledge Online)

對各類數位化的知識，包括業務活動，專案開發，客戶服務相關的文件、軟體、記錄等等分類儲存管理，並依權限提供搜尋下載等服務。知識

庫管理系統架構在資料庫及其搜尋引擎上，透過 Sales Online、Project Online 與 Service Online 提供相關數位化知識的上傳，自動儲存管理、搜尋及下載的服務。目前知識庫管理系統提供下列的分類知識管理：(1) 業務類：產品介紹，競爭廠商資訊，建議書，報價單，採購合約書。(2) 專案類：專案文件：系統分析、系統設計、程式、使用手冊、問題庫。(3) 軟體元件：依用途、程式語言等分類，並提供版本的控管。(4) 範例程式：依程式語言分類，並提供版本的控管。(5) 服務類：維護合約，客服 FAQ。

4. 即時業務支援系統 (Sales Online)

提供業務銷售過程中，客戶相關的人、事、競爭廠商資料與歷史記錄，並提供產品報價等輔助工具，協助業務人員有效達成目標，並透過 Project Online 與 Service Online 即時掌握合約執行狀況與客戶反應，並提供合約執行異常的警示及合約執行檢查點、驗收、收款等關鍵時程的提醒。透過年度銷售計劃、客戶拜訪、進度追蹤與業務活動異常的警示，讓業務主管隨時掌握業務動態，提供第一線業務同仁即時的協助。

5. 客戶服務系統 (Customer Online)

讓客戶視凌群為一虛擬資訊部門 (Virtual IT Department)，不但可透明化的隨時掌握委託凌群承製的專案進度與內容，未來更可透過數位簽章，完成網路化的軟體交付與驗收，可隨時追蹤問題的處理進度，並可透過經驗庫搜尋提高問題判斷與處理的能力，本系統亦提供多國語言介面，

以利跨國業務的推展。

6. 即時決策支援系統 (Management Online)

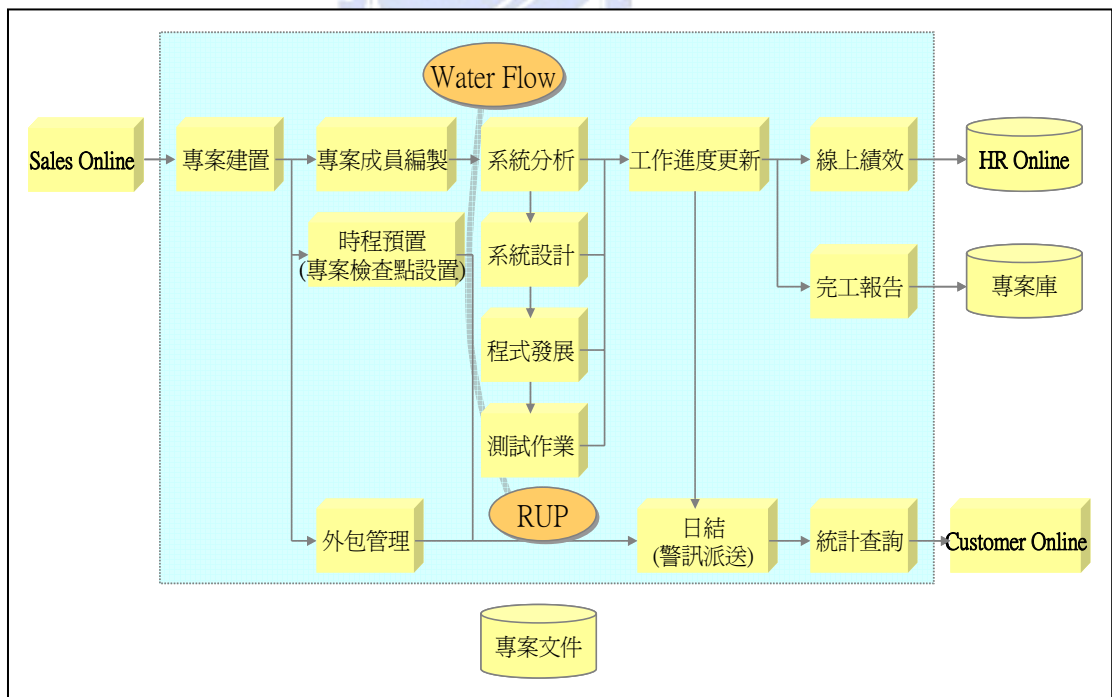
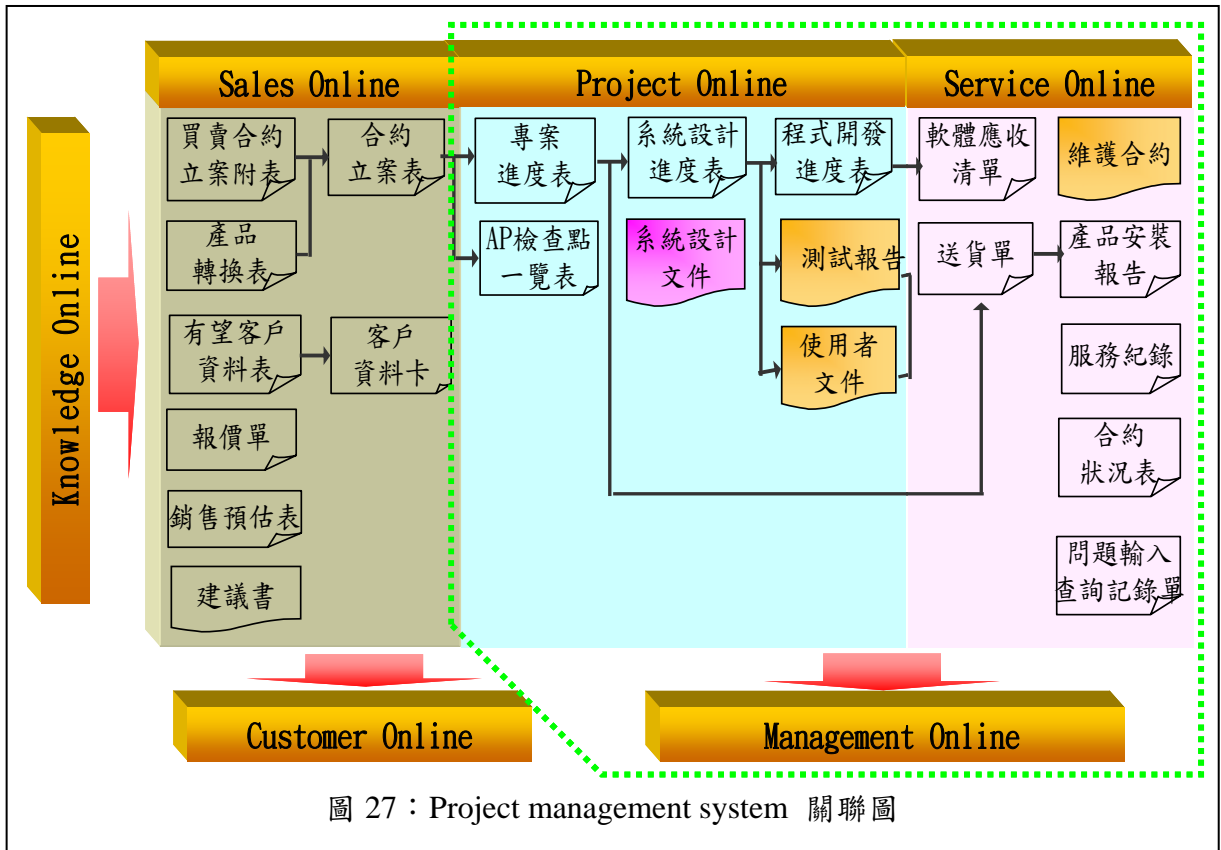
提供主管依據權限，即時掌握整體資源的動態包括：業務動態、業績達成情形、專案執行、客戶服務的異常情形，可即時掌握及處理，並有效提高管理的決策品質，及有效規劃企業的整體資源。

7. 人力資源管理 (HR Online)

即時動態管理員工的技能、專長、可用時間與工作容量。內容包括：員工學經歷，具備的技能及等級；由內外部訓練，專業認證，專案考核等作等級的核定，個人年度技能培訓規劃，配合 Project Online 及 Service Online，依據參與專案式服務的角色與內容，預估參與時間的範圍，動態管理每人可用時間及可用容量，並提供技能、工作地點、可用時間等各種條件配合的搜尋。

8. 協同作業系統 (Partner Online)

為專案管理系統的延伸，將合作伙伴視為專案開發的虛擬團隊 (Virtual Team)，透過角色權限等的設定可將專案管理的功能透過 Partner support system 延伸到虛擬專業團隊的所有成員。協同作業流程由業務服務系統依據中長期有望客戶預估所需產能，到專案成立、派工、測試、結案、維護。全部以線上即時監控，與軟體工廠最有關係的管理系統為專案管理系統 (Project Online) (圖 27 所示)。



Project Online 將專案開發過程中的合約需求、設計文件、變更記錄、使用手冊、程式碼等資料加以分類管理，並提供績效管理、流程管理、知識管理等功能。透過系統的輔助可以使專案發展過程的成本、進度透明化，提高品質（圖 28 所示）。其主要功能如下：

1. **專案組織**：透過 HR Online 尋得最符合專案所需的技能與時間等關條件的人力並付與適當的專案角色。
2. **專案建立**：可透過 Sales Online 或自行建立專案相關基本資料及執行檢查點。
3. **專案進行**：提供兩種軟體開發程序 Water fall (Default) 與 RUP 的自動程序控管，並隨時更新每日專案成員的進度，透過上傳每一階段產出的文件或程式碼集中於知識庫的專案控管，並對進度落後於預定進度提供各式警訊。
4. **專案成本評估與結案報告**：透過專案成員每日工時記錄，累計專案人力成本與彙總其它專案成本，與預估成本作比對並記錄專案執行心得作為經驗分享。
5. **線上績效**：定期及結案時主動提供專案成員績效評估表，提供專案經理有效管理工具，並提供專案成員年度績效及培訓計劃之重要依據。

Project online 在專案不同階段做不同的管理工作，包括：專案的成立與控管、開發程序的選擇、成員進度的掌握、開發文件管理與知識/元

件的引用、異常管理、專案成果的累積、專案績效。可於任一時點瞭解專案開發狀況，知道 Project 的狀態；是否落後、超前、遇到什麼問題，由過去幾週發展紀錄，可以看出發生的問題，去追蹤計畫是否持續推動中。了解 Project 資源投入：金錢、人力資源、間接物料的投入，研發人力資源運用狀況，風險管理，提供每個專案計畫物料成本數據。

6. 文件管理與分享：Project online 將軟體發展中的文件資料儲存在不同的知識庫，並透過權限分享的機制加以管理，軟體發展團隊成員與管理人員可以分享經驗與知識（圖 29 所示）文件管理除了提供相關專案人員也可以納入知識管理的工具 Knowledge online 管制，讓相關人員依權限分享文件。



圖 29：Project online 的文件管理

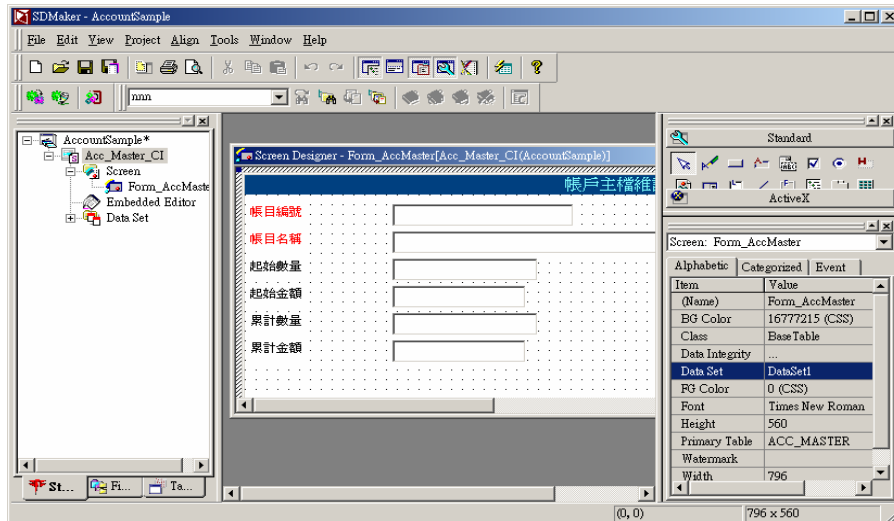


圖 30：SDMaker 設計畫面

個案軟體工廠生產中心使用自行研發的工具 SDMaker (圖 30 所示)，依目的碼 (Object Code) 的不同分為：SDMaker - JSP、SDMaker-VB、SDMaker-ASP，不同版本的工具差別在於程式碼產生器 (Code Generator) 產出的程式碼為 JAVA 或 VB 或 ASP (圖 31 所示)，不同專案再生產線上套用不同的模版，因此在線上派工時必須詳細註明專案名稱、模版等資訊。使用線上派工的原因在於軟體工廠可能在不同地區，如同一醫療專案參與的專案人員分散在台南、台中、台北、大陸西安等地。

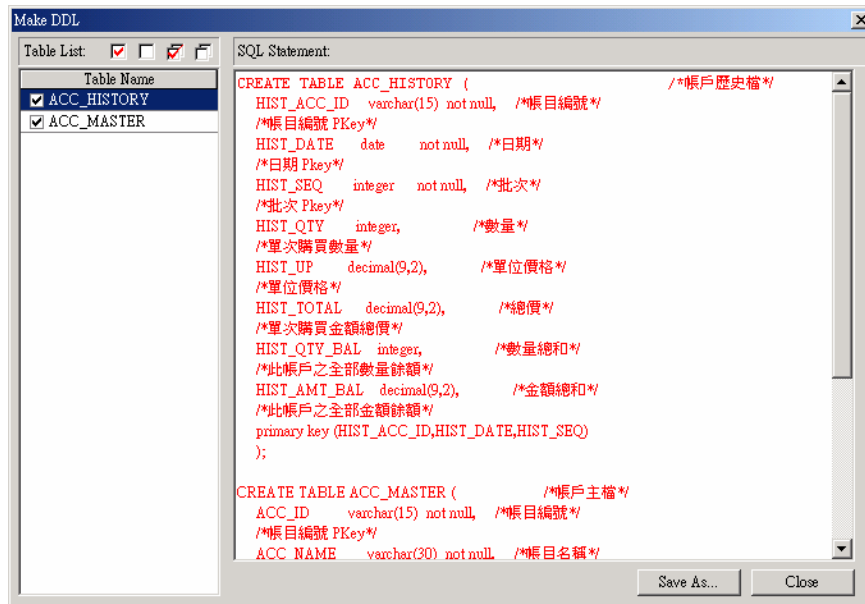


圖 31：SDMaker 程式碼產生器

軟體工廠的兩項主要工作是架構設計與程式碼生產，軟體模版是生產工具 SDMaker 用來產生程式樣式的基本雛形定義，並非所有的程式都可以套用模版，依經驗統計約 60%-70% 可以直接套用，主要為資料輸入類、資料查詢類、報表類、檔案維護類。其他尚有無法套用模版的程式約 30%-40%，如核心程式、線上交易邏輯、工作流程類等。此類程式需遵循標準，標準範例由委員會制定，委員會由跨公司跨部門技術主管組成，定期開會定期公告標準。

4.4 建立績效指標

軟體工廠線上生產可蒐集量化數據績效指標，做為管理參考，指標的設定需和管理目標一致，個案公司的績效指標如：元件再使用率、平均生產力、測試不良率[11]、專案時程提升率，如表 3、表 4 所示。

表 3：量化指標

時 間 績 效 指 標	Nov-89	Dec-89	Jan-90	Feb-90	Mar-90	Apr-90	May-90
	元件的再用率	1.37	1.42	1.79	2.35	2.56	2.7
平均生產力	20	23	24	25	28	29	32
測試不良率	4.12	3.85	3.6	3.2	2.9	2.7	2.4
專案時程提升率	-0.15	-0.13	-0.07	0.02	0.09	0.15	0.17

表 4：指標推算說明

元件的再用率	程式元件被參考次數合計 / 程式支數合計
平均生產力	規格加權後程式總數 / 程式人員總數
測試不良率	不良程式支數 / 程式總支數
專案時程提升率	(預估使用工時 - 實際使用工時) / 預估 使用工時

個案公司其他非量化指標成效如表 5 所示：

表 5：非量化績效指標

1	產品品質獲得改善（缺失降低 10%）
2	程式碼錯誤明顯減少，軟體設計成本降低
3	研發管理品質及成效獲得雙重提升
4	專案預算控制成效顯著
5	專案管理符合預期進度，客戶滿意度明顯上升



五. 結論

5.1 軟體工廠的效能

軟體工廠以製造的生產概念套用在軟體設計與開發流程，並藉由工具的支援達到三項控制目標：品質、成本、時程。個案公司在嘗試工廠生產模式後已逐漸發生效益，歸納軟體工廠的效能改善因素有下列三點：(1) 標準化：使用工具，再使用元件，標準介面，生產管理。(2) 批量化：產能規劃，成本控制。(3) 彈性化：彈性人力資源，外包管理。為了大量生產降低人員的干擾因素，軟體工廠最關鍵的地方為標準化，並使用生產工具和管理工具支援，將軟體生產元件化，累積的再使用元件與軟體模板可以降低成本，在生產過程中收集相關量化數據，做為管理和控制的依據。生產中心的建置是為了大量生產，軟體模版可以大批量生產相關程式，因為標準化流程可以預估產能與成本，利於做中長期產能預估與成本控管。軟體人員的流動率高，因為使用標準工具與流程容易建力協同作業模式，可以彈性運用人力資源，並做好外包管理。

5.2 未來研究方向

標準化製程除了在內部推動以外也要考慮國際標準，1991 年美國卡內基大學美濃 (Mellon) 學院的軟體工程組織 (Software Engineering Institute) 發表的軟體製作能力成熟度模式 (Capability Maturity Model, CMM)[8][9] 是針對軟體產業所定的評估模式，成熟度模式分為三

類，有軟體 CMM (SW-CMM)，系統工程 CMM (SE-CMM)，軟體取得 CMM (SA-CMM)，CMM 的架構將管理流程能力的成熟度分為 5 級，每一級包括以幾個主要的流程區塊 (Key Process Area, KPA)，每一個流程區塊包括該級應有的目標，應具備的執行能力，度量和分析方法。根據軟體工程組織 (SEI) 的調查使用軟體 CMM 的效益有：(1) 生產力提高 35%。(2) 產品上市時間提前 19%。(3) 交貨後發現的瑕疵率降低 39%。(4) 節省的成本與投入改善的成本比為 5：1。因為改善成效顯著，成熟度模式受到軟體大廠的重視，紛紛導入 CMM 改善軟體生產品質。為了將「軟體」、「系統工程」、「軟體取得」三個來源模式整合，2003 年 10 月 SEI 宣布將 CMM 提昇為 CMMI (Capability Maturity Model Integration) 成為目前最新版本。未來研究方向可以朝軟體工廠如何導入 CMMI 評鑑對軟體生產製程及提升軟體品質的效益分析，針對專案的量化分析與管理能力進行各項參數與量化指標的蒐集，以作為量化分析之基礎，進而就量化分析的結果進一步反饋當作專案開發流程改善之依據，並透過 CMMI 模式讓軟體工廠製程滿足國際標準。

參考文獻

- [1] 王凱，「資訊軟體市場發展現況與趨勢」，財團法人資訊工業策進會資訊市場情報中心，民國 94 年 1 月。
- [2] 周樹林，「我國軟體發展與管理能力應用趨勢分析」，財團法人資訊工業策進會資訊市場情報中心，民國 91 年 12 月。
- [3] 施振榮，「深談微笑曲線」，科技產業資訊長協進會（CIO 協進會演講，民國 93 年 9 月 19 日。
- [4] 詹文男，「等待英雄，再造台灣資訊軟體產業」，財團法人資訊工業策進會資訊市場情報中心，民國 92 年 11 月。
- [5] 黃種智，「軟體發展組織再造與軟體發展程序之配合—以凌群電腦為個案研究對象」，國立交通大學高階主管管理學程，碩士論文，民國 90 年 6 月。
- [6] 黃繼弘，「軟體產業導入供應鏈分工合作之機制探討—以企業資源規劃系統水平整合為例」，國立政治大學資訊管理學系碩士班，碩士論文，民國 91 年 8 月。
- [7] 張清峰，「協同軟體研發專案管理之研究—以全球化研究團隊整合為例」國立台灣科技大學工業管理系，碩士論文民國，91 年 8 月。
- [8] 財團法人資訊工業策進會，CMMI 導入指引 V1.0[第二級]，民國 91 年 12 月 31 日。

- [9] 財團法人資訊工業策進會，CMMI 導入指引 V1.0 [第三級]，民國 91 年 12 月 31 日。
- [10] Detlev J. H. 等著，張國鴻譯，數位式競爭-全球軟體公司的制勝策略，台北，天下文化出版社，民國 91 年 10 月。
- [11] 許育誠，軟體設計與品質管理，台北，文魁資訊股份有限公司，民國 92 年。
- [12] 信懷南，不確定年代的專案管理，聯經出版社，民國 90 年 6 月。
- [13] 吳仁和，林信惠著，系統分析與設計，台北，智勝文化，民國 92 年 3 月。
- [14] Standish Group, Chaos - “CHAOS (2004)”, 2004, http://www.standishgroup.com/sample_research/
- [15] Bemer, R.W., “Position papers for Panel Discussion: The Economics of Program Production. In: Morrell, A. J. H. [eds.]: Information Processing 68”, North-Holland, Amsterdam, 1969.
- [16] Lim, N.K., Ang, J.S.K., Pavri, F.N., “Diffusing software-based innovation with a software factory approach for software development”, Management of Innovation and Technology, 2000. ICMIT 2000. Proceedings of the 2000 IEEE International Conference Volume: 2 pp. 549 -555, Nov. 2000

- [17] K. Fujino, “Software factory engineering; today and the future” , Proc. of the Computer Conf.’ 87 Fall Joint, 1987.
- [18] McIlroy, M. D. , “Mass-Produced Software Components In Software Engineering” , Reports on a Conference Sponsored by NATO Science Committee, Brussels, 1969.
- [19] Watts S. Humphrey, “Software and factory paradigm” , Software engineering Journal, September 1991.
- [20] “Software factory principles, architecture and experiments” , IEEE Software, March 1992.
- [21] Cantone, G. , “Software factory: modeling the improvement” , Factory 2000, 1992. ‘Competitive Performance Through Advanced Technology’ ., Third International Conference on [Conf. Publ. No. 359] , 27-29 Jul 1992, pp.124 - 129.
- [22] James R. Johnson, The software factory-managing software development and maintenance, 2nd ed, QED Information Sciences/c1991.
- [23] Cusumano, M. A. , Japan’ s Software Factories, Oxford University Press, 1991.

- [24] Cusumano, M.A., “The Software Factory: A Historical Interpretation” , IEEE Software, March 1989.
- [25] Fernström, C, K-H. Närfelt, and L. Ohlsson, “Software Factory Principles, Architecture, and Experiments” , IEEE Software, March 1992.
- [26] Matsumoto, Y.: SWB System: A Software Factory. In: Hunke. H. [ed.]: Software-Engineering Environments, North-Holland, Amsterdam, 1981.
- [27] Matsumoto, Y.: A Software Factory: An Overall Approach to Software Production, In: Freeman, P.: Software Reusability, IEEE, 1987.
- [28] Michiel van Genuchten, Towards a software factory, Kluwer Academic Publishers, 1992.
- [29] James R. Johnson, The software factory-managing software development and maintenance, 2nd ed, QED Information Sciences/c1991.
- [30] J.F. Nunamaker, et al., “Task force software productivity and the automated software factory” , Proc. of the Twenty-First Annual Hawaii International Conference, Volume: 4, 5-8 Jan.

1988.

- [31] Chao Li, Huaizhang Li, Mingshu Li, “A software factory model based on ISO 9000 and CMM for Chinese small organizations” , Quality Software, 2001 Proceedings Second Asia-Pacific Conference on , 10-11 Dec. 2001 pp.288 - 292.
- [32] Rockwell, R., Gera, M.H., “The Eureka Software Factory CoRe: A Conceptual Reference model for software factories” , Software Engineering Environments Conference, 1993. Proceedings pp. 80-93, July 1993.
- [33] Verrall, M.S, “ESF: the Software Bus” , Architectures for Distributed Development Support Environments, IEE Colloquium on, 4 Nov 1991, pp. 4/1 -4/3.
- [34] Fernström, C., “The Eureka Software Factory: Concepts and accomplishments. In: Lamsweerde, A. and A. Fugetta [eds.]: Proceedings of the 3rd European Software Engineering Conference. Lecture Notes in Computer Science No. 550, Springer-Verlag, 1991.
- [35] Basili, V. R., “The Experience Factory: packaging software experience. Proceedings of the 14th Annual Software Engineering

Workshop, NASA Goddard Space Flight Center, Greenbelt MD 20771, 1989.

- [36] Basili, V.R., G. Caldiera, and G. Canone, “A Reference Architecture for the Component Factory, ACM Transactions on Software Engineering and Methodology, 1992.
- [37] Martin L. Griss, Kevin D. Wentzel, “Hybrid domain-specific kits for a flexible software factory”, Hewlett-Packard Laboratories.
- [38] Mintzberg, H., Structures in Fives: Designing Effective Organizations, Prentice-Hall, 1983.
- [39] Nilsson, EG., “CASE Tools and Software Factories”, In: Steinholz B., A. Sølvsberg, and L. Bergman [eds.]: Lecture Notes on Computer Science. Berlin, Springer-Verlag, 1990.
- [40] Thoreson, S. A., “The automated software development project at McDonnell Aircraft Company” (the Software Factory), Aerospace and Electronics Conference, 1989. NAECON 1989., Proc. of the IEEE 1989 National, 22-26 May 1989, pp 1576 -1580, vol. 4.

- [41] Conn, R. , “Developing software engineers at the C-130J Software Factory” , Software, IEEE, Volume: 19 Issue: 5, Sept.-Oct. 2002 pp. 25 -29.
- [42] Jack Greenfield, Keith Short, “Software factories: assembling applications with patterns, models, frameworks and tools” , Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, October 2003.



個人簡歷

姓名	陳裕昌	性別	男	出生 年月日	民國 46 年 7 月 23 日
學歷	逢甲大學 資訊工程研究所				
經歷	曾任：凌群電腦 系統工程師 凌群電腦 市場處產品 經理 凌群電腦 中區事業部 協理 凌群電腦 電子商務事業部 協理 凌群電腦 應用軟體總處 處長 凌群電腦 金融產品事業部 資深協理				