

---

# 教 學 個 案

---

◎專案管理與資訊系統開發—以聖州企業為例

◎日本式生產管理技術移轉之管理策略—中華汽車  
公司個案



## 專案管理與資訊系統開發 —以聖州企業為例

A Case Study of Forcartex's System Development  
and Project Management

吳仁和 *Jen-Her Wu*

國立中山大學

National Sun Yat-sen University

<sup>1</sup>林三順 San-Shuenn Lin

National Sun Yat-sen University

<sup>2</sup>夏則智 Tse-Chih Hsia

美和護理管理專科學校

MeiHo Junior College

聖州企業（股）公司成立於民國六十八年，總公司設於台北並分別於台中及屏東設立分公司與工廠。該公司屬於家族企業，由母親任董事長，並由兄弟分別擔任總經理及副總經理等要職。公司主要從事汽車配件（如車套、椅套、腳踏版與背墊等）的生產與銷售，產品除了自行生產外，部份直接向其他廠商採購，銷售區域則涵蓋國內外。公司目前之員工約五十人，一年之營業額約八千萬，其中內外銷約各佔一半。

長久以來公司之營運管理與分析等工作全靠人工來處理，因此不但耗費人力，且管理階層無法及時取得公司之營運分析資料（例如公司之目前營利、某訂單之獲利與原料用量、成品庫存及銷售情況等），以有效的幫助其瞭解公司現況及做為決策的依據（例如視各區銷售情況，以調撥庫存成品至缺貨區）。

公司副總經理深感目前同業之競爭壓力及未來企業之擴展需求，若不及時引進資訊系統以支援管理及分析工作，進而提升企業之競爭力，公司將很快的面臨生存危機。因此，乃毅然決定將企業一部份管理工作電腦化，並委託屏東一家小型資訊公司進行資訊系統之開發、軟硬體規劃與採購。

受委託之資訊公司主要是經營硬體銷售兼小型軟體之開發，該資訊公司接受聖州委託後便開始進行需求分析工作。經一段時間之需求瞭解及估算後，該資訊公司預計半年內完成系統建置。雙方在此認知下，便開始進行系統開發之進一步工作，包括系統分析與設計、程式撰寫等。半年時間很快就過去了，然而完成之功能未及三分之一，且程式仍存在有許多錯誤。在此期間雙方仍有頻繁之互動，希望能瞭解問題及解決問題。但資訊公司感到該系統所處理問題之複雜程度並非如當初預估的那麼單純，因此對該系統之開發工作逐漸感到力不從心。

資訊公司在成本壓力及結案遙遙無期情況下，對此案已逐漸失去信心，而與聖州方面溝通之次數愈來愈少。聖州方面也非常體諒資訊公司之窘境，故一再給予資訊公司時間與機會，希望大家能共同將系統建置完成。轉眼一年半過去了，此時雙方都瞭解該系統已無法成功的建置，因此系統開發合約乃正式終止。

電腦化失敗後，公司的作業變得更加混亂，因為原來是完全人工，但現在是人工與電腦化參半。例如在生管方面，庫存及用料部份電腦化，其餘人工作業，因此衍生出部份工作重複與資料不一致等問題。歷經一次失敗的經驗後，副總經理深感資訊公司之能力是系統成敗之關鍵因素之一，但因限於預算，故找大型資訊公司開發之可行性不高（因大公司不願意接小案子）。因此，副總經理便嘗試向學術單位尋求協助，經一段時間之努力後，決定委託中山大學資訊管理學系軟體研發中心（以下簡稱研發中心）重新開發聖州公司之資訊系統。

研發中心與聖州於八十五年三月開始洽談資訊系統開發事宜，聖州方面先指定生產部兩位資深人員負責向研發中心描述上次電腦化之範圍及失敗之情況，並提出新系統須支援之作業需求與報表等。研發中心初步評估該系統在專業知識及資訊技術方面沒問題後，便進一步瞭解各作業之程序、輸入、輸出及相關之運算等。這包括收集現行作業程序規範，各種營運管理與分析報表等，以界定系統範圍，並釐清系統應具備之功能。經過三次需求訪談及

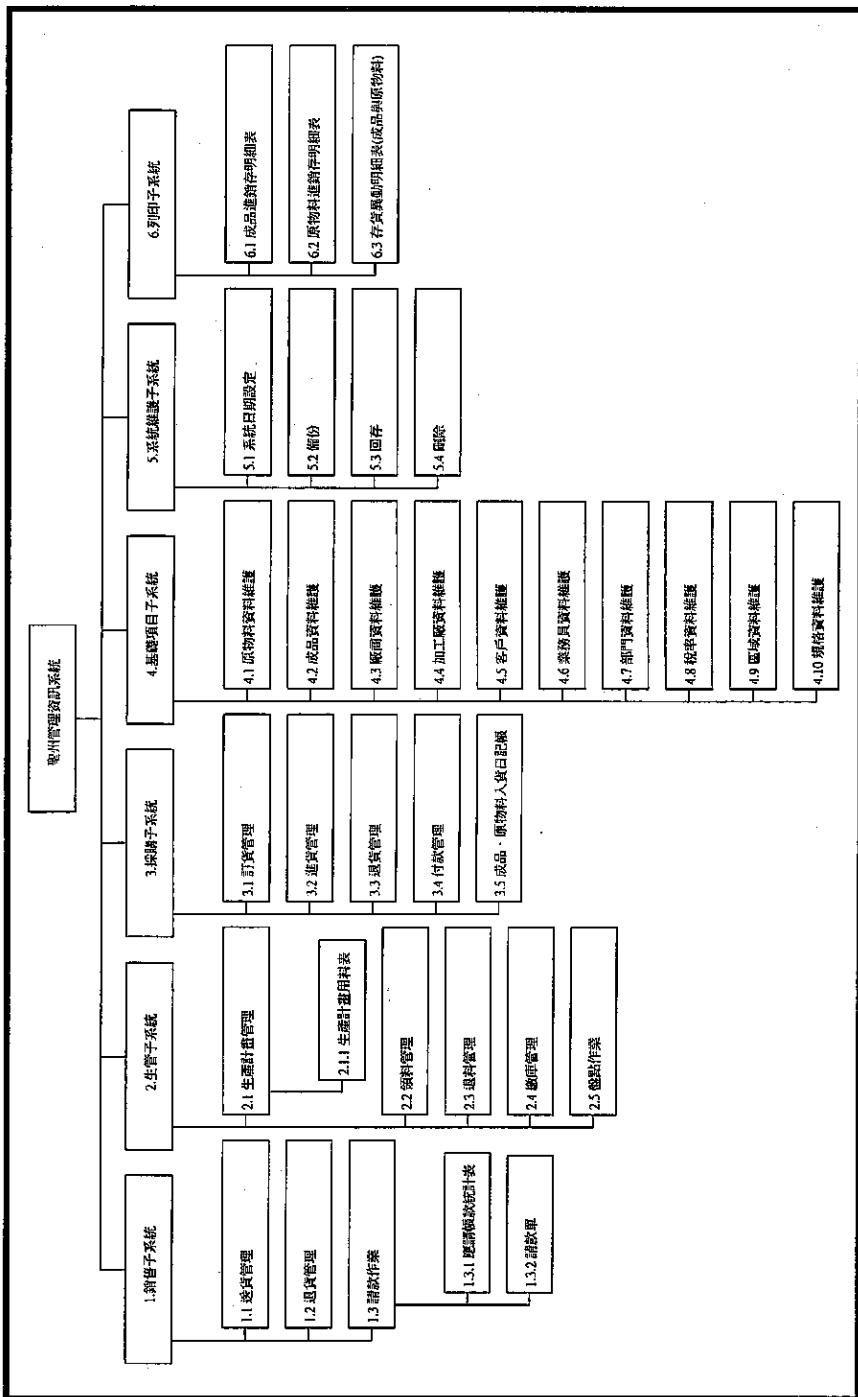
確認後，研發中心整理出聖州之需求概可分為銷售、生管、採購、基本資料維護、營運報表分析與系統維護等六大子系統，各子系統內之作業項目及列印表單詳如圖一。

在軟／硬體規劃方面，因限於公司之預算與人員編制，並考量使用者於資訊管理方面之相關知識，聖州系統乃採兩期進行開發與導入。第一期建議採用單機版，詳細之軟/硬體需求如表一。俟系統導入成功後，第二期再擴充成網路版，並給予使用者有關網路系統管理方面之教育訓練，使能自行操作與維護系統。研發中心整理出上述系統及軟/硬體需求後，便進一步估算所需工時(600 小時)及管銷成本並向聖州報價。研發中心僅承包系統開發，硬體部份由聖州自行採購。雙方先談妥價格，並約定系統開發工作預計於八十五年七月正式動工，且於半年內完成系統建置及測試。

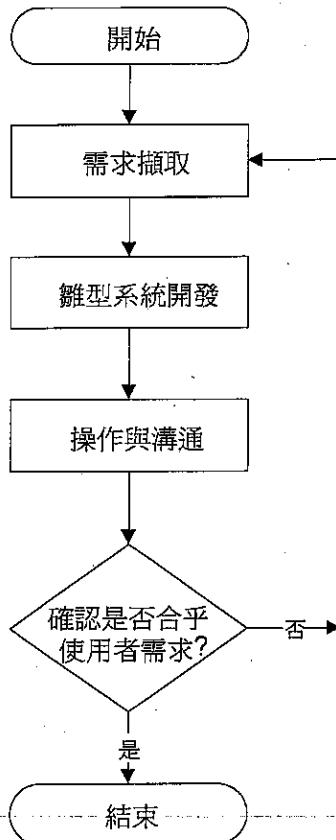
表一 軟/硬體需求

項 目		備 註
軟體	1.Delphi 2.0 Client/Server 版	系統開發工具
	2.Paradox	資料庫
	3.Windows 95	系統執行環境
硬體	1.IBM 相容 PC(Pentium CPU，16M RAM 以上)	工作站
	2.UPS	不斷電系統
	3.點矩陣 24 針印表機	列印用
	4.可讀寫光碟機(MO 230MB)	資料備份用

在初步需求擷取與確認過程中，專案經理已感覺到使用者對需求表達非常模糊，甚至有不知如何表達或不知其需求是什麼之情況。在此情況下，顯然傳統之系統發展生命週期法 ( System Development Life Cycle , SDLC ) 並不適用於此案，因為應用該方法之前提是使用者需求很明確，且系統開發者在系統開發初期便清楚掌握所有使用者的需求 ( 包括作業程序、輸出/入資料之屬性與格式、資料間之關係及相關之運算等 ) ，然後循序的進行系統開發之各項工作 ( 例如系統分析與設計、程式編輯與測試等 ) 。因此，專案經理乃決定聖州管理系統之開發採用演進式雛型法且依基本資料維護、銷售、生管、採購與營運分析等先後順序將系統逐步導入。



離型法之應用是從雙方（使用者與系統開發者）需求最確定之部份（並不需要一開始便全盤瞭解使用者之需求，故與 SDLC 不同），或者是優先順序較高之部份，先進行系統開發之各項工作（所需工作與 SDLC 同），並很快的把離型開發出來，且安裝在使用者之環境供操作與練習。雙方再以離型做為需求溝通與確認之工具，若離型系統之各項功能與使用者之需求有任何出入或不對，使用者可藉由離型系統清楚的指出錯誤及說明訂正方式等。系統開發者乃依此修正系統，然後將修正過之系統讓使用者再度操作與確認。如此反覆進行，直到符合使用者需求為止。接著再對另一部份之需求，依相同程序反覆進行系統開發，直到所有使用者需求均完成為止。演進式離型法之系統開發程序詳如圖二。



圖二 演進式離型法之系統開發程序

應用離型法於系統開發，一方面使用者可藉由離型之練習，因而瞭解資訊系統能做與不能做之工作，可以激發其新需求之產生及讓需求表達更具

體。另一方面，系統開發者亦可藉由利用雛型系統與使用者溝通的過程中，學習有關系統所處理問題領域之專業知識，更瞭解使用者真正的需求，或藉以引導使用者具體的表達需求等。

在聖州之個案，由於使用者對各部份之表達都是一樣的模糊，因此聖州系統開發與導入由系統中資料流之順序來決定各子系統導入之先後順序。例如，基本資料維護是本系統所有原始資料之來源，因此它需要先被建立。其次，銷售子系統處理有關訂貨、送貨、退貨及請款等作業，該作業是公司營運的第一線，故其優先順序應僅次於基本資料維護。同理，接下來之優先順序依序為生管、採購與營運分析等，因為銷售資料可作為生管之依據；同樣的，生管之資料是採購處理之重要輸入。俟各子系統均上線後，系統中才有完整的營運資料，因此最後才進行營運分析。

聖州系統如期開始動工，研發中心除由一位老師擔任專案經理，另由兩位學生分別任分析師與程式設計師等職。專案經理負責初期之需求擷取、估價及進度管控等；分析師負責進一步之系統分析與設計，安排系統開發工作，與程式設計師共同參與程式編輯。

系統開發之初，部份之基本資料維護功能很快的完成，並交由使用者鍵入資料。但使用者由於本身業務繁忙，資料鍵打的時間有限，故進度非常緩慢。在此期間銷售子系統之開發工作已開始進行，並於兩週內完成雛型，再交由使用者操作。此時需求表達與確認方面之問題已開始浮現，例如送貨處理之課稅類別可分為未稅、稅內含及稅外加三種，各有其不同的計算方式。需求擷取是經由專案經理與使用者溝通後再傳遞給分析師，專案經理在擷取第一手資料時，是由使用者口述，專案經理記錄並整理出公式，再交由使用者確認。過程中使用者可能業務太忙，無心或無法瞭解公式之表達方式，因此雖然使用者確認了公式，但稅內含之計算方式仍然錯誤。

此外，使用者遺漏需求之問題亦層出不窮，例如使用者一開始並未告知銷售之商品除了成品（主要的）外，亦可能會有少量之原物料，故銷售子系統雖已完成但仍必須再修改以符合實際作業。此類之程式修改幅度遠比上述計算方式之修改來得大，因為涉及資料庫綱目與程式碼之修正。

上述需求變更之修改情況已嚴重的影響系統開發效率、完成時程與系統開發成本，因為成本之壓力，專案經理已開始思考應用雛型法所帶來的一些問題。例如，從需求擷取與確認之方面來看，雛型法之應用的確可讓使用者

及早操作雛型系統，因而可儘早發現問題及解決問題；另一方面亦可幫助使用者需求表達，及具體的瞭解資訊系統之能力與限制（尤其是無資訊系統相關背景的）。但因需求改變或新增，造成系統反覆修改之成本應由誰來負擔呢？

專案經理思索一段時間後決定暫時先自行吸收，並從專案管理面著手，希望能更實際的瞭解開發過程所發生過之問題與更有效的擷取使用者之需求，以提升研發中心之系統開發效率。這方面之措施包括調整分析師之部份工作，使其直接面對客戶，以縮短原來需求傳遞經由使用者、專案經理、分析師至程式設計師之路徑。另外，利用甘特圖（Gantt Chart）管理專案計畫，並讓參與者均瞭解專案之工作與時程。專案經理每週依甘特圖之計畫分配工作給分析師與程式設計師，並要求每天依實際之進度及所遭遇之問題填寫工作日誌。

另一方面，要求使用者提出新需求時，必須有清楚且正式的文件描述，例如報表需求必須要畫出格式，填好項目及可能之項目計算公式等。此方式可幫助使用者較深入的思考其需求，以減少需求表達錯誤與遺漏情況的發生，又可以幫助系統開發人員瞭解需求，進而提升需求擷取及確認之效率與效度。除了上述兩方面之改善措施外，該專案每週定期檢討，以瞭解進度、處理所發生之各種問題及相互交換心得等。

系統開發進行至七月底，雖然不如預期中的順利，但至少基本項目維護與銷售子系統之功能（請參考圖一）已大部分完成。使用者用了雛型系統後，從中學習到資訊系統應該可以在更多的方面提供有力的支援，尤其在分析方面。因此，副總經理便指定由財務部陳主任於七月底擔任聖州方面的總召集人，並首次正式提出新增各種營運分析之需求。

研發中心經與陳主任溝通後（八月五日），整理出新增之功能包括登帳作業、銷售、生產與採購之營運分析功能及分析報表之產生等（詳如表二）。研發中心經評估後，認為上述之新增對系統修改的幅度並不大，因此接受聖州所提出之需求新增，並根據分析師所估算之工時酌收工本費。

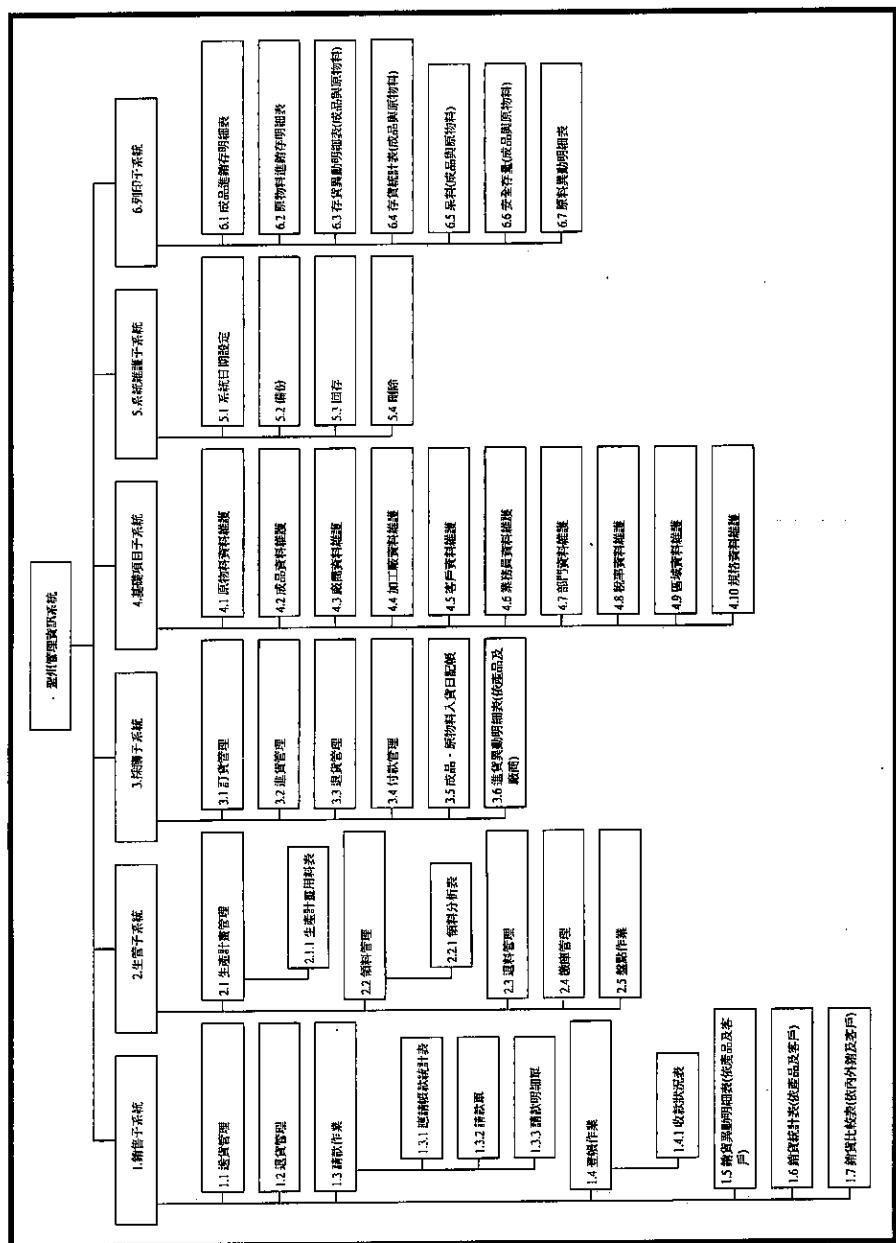
表二 第一次要求增修之系統功能

1.銷售子系統
1.4 登帳作業（新增）
1.4.1 收款狀況表（新增）
1.5 銷貨異動明細表(依產品,客戶各一)（新增）
1.6 銷貨比較表(依內外銷,客戶各一)（新增）
3.採購子系統
3.6 進貨異動明細表(依產品,廠商各一)（新增）
6.列印子系統
6.5 呆料(成品、原物料)（新增）
6.6 安全存量(成品、原物料)（新增）

雙方談妥後，聖州系統便依更新後之功能進行系統開發。八月下旬，銷售與生產子系統已大致完成並交由使用者操作與測試，此時採購子系統之開發工作也已開始進行。使用者用了一段時間後，對離型系統之能力非常滿意，但一些新需求又陸續浮現，例如在生產子系統之領料、退料與繳庫單等都必須分別註明單位及類別，各分析報表要增加查詢條件等。此外，又於八月底，第二次要求修改與新增營運分析之報表及其內容。例如，新增請款明細單與銷貨統計表；修改銷售比較表，將原來報表分成內、外銷，且內銷分為北、中、南區，外銷分為直接、間接與外購等。第二次增修之功能及報表摘述如表三，修改後完整之聖州系統功能表如圖三。

表三 第二次要求增修之系統功能

1.銷售子系統
1.3.3 請款明細單（新增）
1.6 銷貨統計表(依產品,客戶各一)（新增）
1.7 銷貨比較表(依內外銷,客戶各一)（修改報表內容）
6.列印子系統
6.4 存貨統計表(成品、原物料)（修改報表內容）



圖三 聖州系統功能表

雛型系統已經過了多次修改，專案經理心想該系統應已趨穩定與成熟，因為目前之系統功能應已相當完整且每張報表之項目、內容及格式都經過陳主任確認過了。但事實不然，於九月初陳主任又來電第三度要求將營運分析及在各分析報表（例如存貨統計表、銷貨比較表等）增加排序與百分比等功能，各存貨及銷貨統計表增加”毛利”項目等。

專案經理覺得需求之新增與修改已不能再如此漫無止境的下去了，因為如此下去，不但系統不知何時才能完成，且會讓系統開發者失去信心，產生嚴重的挫折感。因此，專案經理便緊急與陳主任溝通，並說服陳主任第一期系統開發之需求增修到此為止，所有之其他異動都在第二期時一併考量。

應用雛型法於系統開發，雖然開發人員與使用者可以藉由雛型操作及溝通以提升需求擷取與確認之效率與效度，但也引發使用者經常變更需求的情況，使得系統必需頻繁的修改以符合使用者需求。但是，修改之成本如何處理？專案經理也一直為合理的估算需求增修之成本所困，因為每一次修改，系統開發人員必須花費許多時間與使用者溝通，把使用者所提之需求配合原雛型系統重新做整體考量，然後才著手進行系統的修改工作。這些溝通和重新分析與設計之時間及成本每次都必須付出，但對系統開發人員而言卻又難以估算。

另外，因需求增修而導致系統修改之工作量評估上，使用者常無法瞭解為何修改會如此困難或需要這麼多的工時。另一方面，系統開發人員亦往往低估需求修改。其背後所隱藏的複雜性，因此時常低估工時。以上述生產子系統之修改為例，研發中心本以為註明單位及類別應不會有太多工作量，但實際修改時才發現除了程式與資料庫綱目之修改外，還需要修改領料、退料、繳庫間各程式模組之協調關係等，實際所需之處理工時比原先估計還需多出約 50%。

雛型系統從七月動工以來，研發中心除了面對著時間及成本之壓力外，使用者頻繁的額外服務要求亦讓研發中心感到非常困擾。對本案而言，系統開發本應僅限於軟體之開發、雛型系統問題之諮詢與系統操作之教育訓練等，但使用者不管軟／硬體方面之任何問題，皆要求研發中心提供支援。例如，電腦故障與系統當機的排除，Windows 95 環境之重新安裝，印表機更新後的驅動程式調整，系統網路連線之整體規劃、備份諮詢服務等。在系統開發的過程中，上述問題亦耗費研發中心許多的人力資源及成本。

由於上述各因素，因此系統開發進度便節節落後。依專案預定進度，在十月中旬時系統功能應可全數建置完成，並全面在使用者之環境進行系統測試及教育訓練。但專案之進行並不如預期的順利，在十月中雖然大架構均已完成，但類似之小幅需求修改仍不斷的發生。研發中心在儘速結案的大原則下，除了加緊系統的開發速度外，另一方面使用者需求修改之部份亦僅接受不影響其它模組之程式碼及資料庫者。因此，聖州雛型系統於八十五年十一月初正式進入測試階段，並於八十六年三月系統測試及修改完成。

經歷了大約九個月的時間，系統終於完成了。在整個演進式雛型系統開發的過程中，研發中心詳實記錄了每一階段之活動、工作項目與工時（如表四）。由表四的統計發現聖州資訊系統總共耗費了 898 個工時，與在規劃階段所預估的 600 工時相差了 298 工時，也就是大約低估  $1/3$  之工時。是什麼原因造成延誤？

研發中心經過數次檢討及整理專案進行之工作日誌等，以找出可能延誤工時的原因。這些原因大概可歸類為研發中心及客戶所引起，分別整理如下：

### 研發中心方面造成之工時延誤原因

1. 低估系統本身的複雜度
2. 低估與客戶溝通的時間

由表四整理後，可得知研發中心對聖州的訪談與溝通連繫共花費 147 個工時，約佔總工時之 16.37%。

3. 低估系統分析及資料庫設計時間
4. 低估系統整合、測試的時間
5. 低估解決客戶問題的時間
6. 不熟悉系統開發工具

這包括 Delphi 2.0 工具之資料庫與中文相容性的限制、內建之資料字典難以維護等問題所造成。

## 聖州方面造成之工時延誤原因

1. 內部需求不統一，經常變動
2. 需求未能明確描述及一次全部完整的提出（例如報表格式、項目內容及計算公式等）

上述兩個原因即造成了系統兩次的增修（內容如表二與表三），此兩次重大的系統變更共使用了 118 個工時，約佔總工時之 13.14%。

3. 常要求系統之外的需求

使用者不管軟／硬體方面之任何問題，均要求研發中心提供支援。這些額外之需求經由記錄之整理發現總共支援 42.5 工時，約佔總工時之 4.73%。

**表四 聖州資訊系統各開發階段之工作項目與工時**

階 段	活 動	工作項目	工 時
規劃階段	需求定義	訪 談	30.0
		討 論	20.0
		文件製作	4.0
		簡 報	8.0
		小 計： 62.0 (6.90%)	
雛型發展階段	需求分析	訪 談	20.0
		討 論	40.0
		文件製作	8.0
		小 計： 68.0 (7.57%)	
	系統分析與設計	訪 談	30.0
		討 論	41.5
		文件製作	12.0
		小 計： 83.5 (9.30%)	
使用者系統建置	雛型開發與測試	程式撰寫與測試	450.5
		討 論	36.0
		查閱技術文件	13.0
		小 計： 499.5 (55.62%)	
	系統安裝	39.5	
	教育訓練	24.0	
	文件製作	12.0	
	溝通與連繫	30.0	
	小 計： 105.5 (11.57%)		

階 段	活 動	工作項目	工 時
維護階段		系統維護	42.5
		溝通與連繫	37.0
		小 計	79.5 (8.85%)
總計工時： 898.0 (100.00%)			

在整個系統開發過程中，研發中心感受到應用演進式雛型法於系統開發之優點與缺點。以下分別將本個案中所碰到之優點、缺點與建議摘述如下：

### 優 點

#### 1. 提供良好的溝通媒體及可激發需求靈感

因為僅從一部份之需求先著手開發系統，故使用者很快便可操作系統，並藉以具體的指出與需求不符、不足或不對之處。又由於不斷的操作、練習雛型，故使用者可激發許多新的、有用的需求。當然，開發人員亦可從中學習該領域之專業知識。因此，對需求的產生與溝通有很大的幫助。

#### 2. 提高使用者滿意度及降低系統導入之阻力

雛型系統是不斷的根據使用者之意見修正及演進，故所開發出之系統功能較能符合使用者之需求，有部份甚至會高出其預期；若有不足處，也將因為使用者之全程參與，故較能體諒及瞭解其中之困難。另外，由於使用者很早就接觸雛型系統，故系統之導入，使用者有較長的時間準備及調適，因此導入之阻力也大幅降低。

#### 3. 降低系統開發失敗所造成財物損失之風險

因為雙方在系統開發過程之密切合作，故該系統是否能成功，雙方都能洞察在先，因此可儘早採取必要的措施，以防止雙方更多的財力損失及其他傷害。

---

### 缺 點

#### 1. 使用者負擔加重

在系統開發過程，因為使用者本來就有其例行的工作需處理，再加上需要其高度的參與，故常忙不過來，而影響雙方工作之配合。因此，在系統

開發期間，應減少使用者例行之工作量，使期能更專心的參與系統開發及導入之工作。

## 2. 專案管理困難

在系統開發的過程中，因為使用者需求變更以及新需求不斷的提出，專案經理必須經常的召開會議來檢討問題、修改計畫、安排工作及稽核專案進度等，以避免與原先估計時程產生過多誤差。此外，『人』的問題也是離型法的開發過程中常會面臨的。因為需求的增修，系統開發人員必須經常與使用者溝通以瞭解問題及需求，但如果使用者對同一問題的看法不同，則系統的同一功能必須反覆修改，而造成人力及時間無謂的浪費，這些問題都可能影響系統開發人員的士氣及情緒。因此，使用者需求之表達與確認必須要有一套正規之程序與文件讓雙方共同遵循，以提升溝通之效率及釐清責任之歸屬等。

## 3. 人力浪費及系統品質較差

在系統分析與設計工作上，因為無法一開始便對全部需求通盤考量，再加上系統的反覆修改，故不但浪費時間修改程式碼、資料庫與測試等，修改的難度與複雜度亦高，且嚴重影響到系統的品質（例如系統穩定度、程式碼的結構性與一致性）。以資料庫設計為例，若在系統開發過程中反覆修改（例如加入或修改的資料欄位與結構），則已經撰寫好的程式必須修改以配合新資料庫格式，並增加系統測試的負荷。

系統之修改，有些是來自於使用者與系統分析師溝通時產生的認知偏差，也有部份是系統分析師與程式設計師間傳遞訊息時的謬誤。因此，應用離型法，系統開發人員（包括專案經理、系統分析師、程式設計師）必須要有良好的問題領域知識及溝通技巧，並且需要建立起一套與使用者溝通的模式與默契。此外，如何引導使用者能在系統開發時能儘早提出完整的需求，並讓其確實瞭解需求變更後可能造成之成本浪費，對減少系統之修改會有極大的幫助。

## 問題討論

1. 若企業欲進行電腦化，但使用者對其需求常無法清楚表達且在系統開發過程中亦常有新需求提出，面對此種狀況，企業與資訊公司一開始

應如何訂定系統開發之價格？

2. 雜型法適用於使用者需求不明確之情況下的系統開發，假設你是專案經理，你會如何分配成員之工作、工時及評估成員之效率？如何管控該案之進度？相較於 SDLC，雛型法在專案管理上有無差異？
3. 應用雛型法於系統開發，使用者與開發人員是以雛型作為雙方需求溝中之媒介，系統開發人員依使用者所提之需求增修意見修改雛型系統，如此反覆進行，直到使用者滿意為止。但需求之增修常需大幅修改與其相關之其他子系統之程式碼，而導致大量成本之浪費，請問雙方應如何規範（例如需求修改、確認及責任歸屬等），以減少雛型修改之次數？此外，如何決定雛型修改之次數，並能兼顧使用者滿意度及反覆修改之成本？
4. 使用者或系統開發人員常忽略使用者需求修改或新增，其背後所隱藏的複雜性（例如可能需修改原有資料庫、程式碼或引發技術問題），而導致低估系統修改所需之時間與人力成本，請問應如何考量此種成本？
5. 當組織有資訊系統需外包時，如何評選外包軟體公司？評估之準則為何？另一方面，對軟體公司而言，應如何決定是否承接某組織失敗之系統開發案？考量因素為何？
6. 在本個案中是由資料流來決定各子系統開發與導入之優先順序，依序為基本資料維護、銷售、生管、採購與營運分析。假設你是專案經理並決定以雛型法開發該系統，你會如何決定各子系統開發之優先順序？此優先順序是否會因不同的系統開發方法而有所不同？

## 誌 謝

本個案在中山大學資訊管理學系軟體研發中心進行系統研發及個案撰寫，因為全員的支持及辛勤參與，本案才能順利完成，特此致謝。

