

2010 TOPCO 崇越論文大賞

論文題目：

設備維護流程之自動化管理—使用系統
分析方法建構網路程式系統

報名編號： F0036

設備維護流程之自動化管理—使用系統分析方法建構網路程式系統

摘要

在半導體製造業中，關於機器設備的維修皆須要有完備且有系統的記錄，這不但是可顯著改善產品的製造品質，而且可大幅的提升效率。本研究首重產品異常的追蹤系統，經驗顯示，產品的異常不少是由於機器設備的異常所造成。所以，當發現產品品質不良時，重點之一是須追溯機器設備的維修紀錄，以便追查產品的異常原因。在一般傳統產業，甚至於高科技產業，機器設備的叫修紀錄、或產品換線的紀錄更改，大都是以紙本來紀錄。不可否認的，紙本紀錄耗時，容易出差錯，字跡可能不清楚、塗改、或紀錄不確實與不完整等問題，更甚而可能有人為竄改之情事發生；換言之，紙本作業是無效率的，品質低的，且難以提升生產力。執是之故，本研究導入現代化的網頁技術，運用網路程式設計開發出設備維護流程之自動化管理，以達到使用者容易，且管理者可以很有效能的系統化管理。由於使用者可以即時的紀錄、查詢與更新，管理者可及時與遠端管理，實證研究顯示，設備維修時間已明顯的節省，管理的績效也因而提升了。

由於作者本身在機器設備維修系統有很多年的經驗，因此以使用者的角度來進行程式的設計與系統的建構，經過兩年的程式學習與上市半導體封裝科技公司的實務演練及系統測試，加上系統分析與設計及 IT 技術，本研究成功的開發了一套網路設備簽修系統，使設備的簽修系統化和網路化。當機器設備故障時，使用者可透過網路的連結即可以叫修，登錄設備故障新增一筆記錄，而維修人員也可透過網路連結，立即得知設備故障訊息，並於最短時間前往維修。目前，作者已成功的開發完成並實際在 50 台機器設備與 60 人的部門進行作業。在此實際案例，設備的維修工作在簽修系統上線之後，已大幅度的減輕作業員和維修人員資料填寫的負擔，並發揮了網路的功效；設備的管理者透過網路即可以監控設備的維修狀況，以及當產品發生異常時，透過網路即可以立即得到設備的維修紀錄。

關鍵字：產品的異常，機器設備的維修，簽修系統，網路程式

The Automated Management for Equipment Maintenance Process— Using System Analysis Method to Construct Web Programming System

ABSTRACT

In the semiconductor manufacturing industry, regarding equipment maintenance, it requires integral and systematic records, which not only can improve manufacturing quality for products, but also enormously enhance efficiency. This study primarily focused on tracing system for product abnormalities. Experience indicates that many product abnormalities resulted from machinery and equipment abnormalities. Hence, upon the defective products being found, the vital point is to trace the maintenance records of machinery and equipment so that the reason(s) for product abnormalities can be found. In general traditional industry, or even high-tech industry, on call repair and maintenance and product line change records are mostly paper-based records. There is no denial that paper records are time-consuming, easier to incur errors, unclear and altered in handwritings, or record incorrect and incomplete. Even worse, records can be likely tampered with. In other words, paper work is inefficient, low-quality, and hard to enhance productivity. Thus, the current research introduces modern Web page technology, and uses Web programming to develop the automated management for machinery and equipment maintenance process. In doing so, user-friendly and interactive operation can be achieved, and managers can carry out effectively systematic management. Because users can record, query, and update in real time, managers can manage in time and remotely. As this empirical study reveals, time for equipment maintenance has significantly reduced to certain extent, and management performance has also been raised.

The author with many years' work experience in equipment maintenance system can stand in users' position to implement programming design as well as system construction. Through two years' programming learning and empirical practicing and system testing in a listed company of semiconductor electronics, plus system analysis and design as well as IT technology, the current research has successfully developed a networking sign-repair system via Web function to enable the systemization and networking of equipment sign and repair. As machine or equipment broke down, user can call for repair via Web connectivity, record breakdown and add a record of breakdown, and repairman can also via Web connectivity to get the message of equipment breakdown and go for repair within the shortest time. Currently, the author has successfully developed and physically put the system into practice with 50 sets of machinery and equipment and 60 people. For this real case, the work of machinery and equipment maintenance after the launch of sign-repair system has enormously reduced the burden of line employees and repairmen's data filling that has exhibited the networking performance. The managers monitoring equipments can supervise the situation of machinery and equipment maintenance in real time and instantly obtain the records of maintenance via Web or Internet.

Keywords: Product Abnormalities, Machinery and Equipment Maintenance Process, Sign-Repair System, Web Programming

目 錄

摘要	II
ABSTRACT	III
目 錄	IV
表目錄	V
圖目錄	V
第 1 章 緒 論	1
1.1 介紹	1
1.2 研究背景與動機	1
1.3 研究目的	2
1.4 本論文組織	3
1.5 本章結論	3
第 2 章 文獻回顧與探討	4
2.1 介紹	4
2.2 資訊系統設計研發的發展和演變	4
2.3 網際網路應用組成	5
2.4 系統分析的方法	6
2.4.1 系統開發模式的演進	6
2.4.2 瀑布模式	6
2.4.3 雛形模式	9
2.5 本章結論	9
第 3 章 研究方法及個案應用	10
3.1 方法介紹	10
3.2 系統開發生命週期模式	10
3.2.1 系統規畫	10
3.2.2 系統分析	12
3.2.3 系統設計	13
3.3 本章結論	14
第 4 章 實證分析	15
4.1 介紹	15
4.2 程式流程	15
4.3 程式內容	16
4.4 本章結論	20
第 5 章 結論與建議	21
5.1 管理者的迷思	21
5.2 研究結論	22
附錄 A1	52
附錄 A2	53
附錄 B1	54
附錄 B2	55
參考文獻	56

表目錄

表 3-1 硬體規格表	14
表 3-2 使用軟體表	14

圖目錄

圖 2-1 網際網路應用組成圖	5
圖 2-2 瀑布模式	7
圖 2-3 系統開發生命週期 資料來源：HOFFER, ET AL. (2008).....	8
圖 2-4 雛形模式	9
圖 3-1 系統發展生命週期	10
圖 3-2 機故登記表之實際圖片	11
圖 3-3 機器維護表之實際圖片	11
圖 3-4 簽修系統之使用案例圖	12
圖 3-5 須求之資料庫表單關聯圖	13
圖 4-1 功能設計流程圖	15
圖 4-2 設備故障叫修畫面	16
圖 4-3 簽修系統主畫面	16
圖 4-4 維修編輯畫面	17
圖 4-5 維修開始	17
圖 4-6 維修完成圖	18
圖 4-7 維修內容欄位畫面	18
圖 4-8 維修是否完成之頁面	18
圖 4-9 無設備故障畫面	19
圖 4-10 機器維護表	19
圖 4-11 以機台編號分類資料夾	19
圖 4-12 依機台編號列出維修紀錄	20
圖 4-13 依類別搜尋	20

第 1 章 緒 論

1.1 介紹

在本章中將依序提到本次研究的背景、動機與目的，關於研究的背景為研究國內某知名半導體封測廠生產和維修部門所遇到關於機器設備故障，叫修和維修管理的問題。此問題由來已久，大家習慣的依著規範作業，但因時代和科技的改變，我們面對這些問題應該要創新和用新的方法來管理，所以針對這些所遇到的問題，使我們產生興趣，因此本研究的動機為研究是否可以用很簡單和很方便的系統來完成設備維修的簽修動作以及文件紀錄保存及查詢。

關於本研究的目的，除了改善問題外，並探討改善的工具如何取得，系統如何建立，分析目前系統取得的方法，最後將以最低成本的方式來完成系統的建立。此方法是一種創新，結合使用者就是設計者的方式並透過實際學習來達成。本研究從主題確定，歷經兩年時間完成相關課程的學習並學以致用，最後完成系統的建立，不但達成管理問題的改善更提升管理績效，達到事半功倍的效果，在管理問題的解決應用上是一個非常成功的研究案例。

1.2 研究背景與動機

在製造業中，機器設備的保養修理皆須要有作業記錄，機器設備的數量，若不過 10 台，則以紙本紀錄，這是傳統的作法也是簡單的作法。機器設備的數量在大型企業，例如半導體封裝業，單一站別，少則 50 台以上，多則數百台，若是以紙本紀錄，則會有以下問題：

儲存和追查的問題：龐大的紙本紀錄。在 ISO9002 中規定，機器設備維修保養作業紀錄的保存年限為 3 年，因此紙本紀錄須按年按月的存放在儲存室中，以便日後追查。龐大的紙本紀錄，須具備儲存室來儲存，同時數量多時，在尋找上，會發生耗時尋找的問題。

填寫的問題：由於作業繁忙，當設備故障時，現廠操作機器設備的作業人員未必能真正了解設備故障的原因，所以填寫叫修單時，故障原因欄，往往只填寫“機故”兩個字。機故代表機器故障，而如此的填法，並無法真正描述設備故障的情形。

筆跡的問題：填寫紙本紀錄，由於每個人的筆跡並不一樣，有的字跡工整，有的則字跡潦草，尤其是工廠內有大批的外籍人員，英文字跡並不工整，在識別上有困難度。

填寫效率的問題：通常維修紀錄的填寫，須要填寫很多資料，時間的填寫也

不止一次，有叫修日期時間、開始日期時間、完成日期時間等，其他如故障情形，修護情形，再加上流程卡號，客戶的批號等將會造成耗時填寫，以及紀錄填寫不完整的問題。時間對於設備的產能是很重要的，如果能簡化維修人員紀錄的填寫，花更多的時間在設備的維護上，將有助於產能的增加。

設備的維修紀錄，除以上問題外，另外就是紀錄價值性問題。經年累月的填寫紀錄，而紀錄本身除了追溯的意義外，若無其他附加價值，則紀錄的效用及價值不大。如何讓維修紀錄轉化成有用的資料，進而利用它，例如維修紀錄的動作，並不只是一筆紀錄，同時變成設備的監控，即時的處理等，這是另外一個思考的方向，若能將紀錄的動作變成除了是紀錄的性質外，更進一步成為訊息的傳遞，則將可以把管理的工作簡化和更有時效的掌控現場的管理工作。

1.3 研究目的

軟體的應用，取代紙本並進而系統化，所使用的方法不外乎兩種，第一種為委外(outsourcing) 向軟體設計公司例如 IBM 等，購買現成的軟體或者請其代工設計符合使用者環境的軟體 (周坤約，2003)。應用程式的設計和維護費用，少則數十萬，多則數百萬，此為設計軟體不發達時，企業所採取的策略。而除了如此龐大的費用外，另外一個產生的問題則為程式須要改變或增加功能時，須仰賴軟體設計公司，且被軟體設計公控制住，不容易有改變或增加新的功能。因為過了保固期，新增功能或維護就須另外收費，且費用也不便宜。第二，另外一種方法為自行設計軟體 (in-house)。企業為減少成本的支出，尤其在大公司必定設有 MIS (Management Information system) 部門，負責系統的設計與維護。公司的系統主要為 ERP 系統以及網路系統。MIS 部門負責研發公司所須要的系統程式以及日常的系統維護。有了 MIS 部門，公司在 E 化方面就可以省下不少費用且須要增加或改變功能時，可以節省成本和增加時效。

大公司公所具備的 MIS 部門，原則上所設計出的應用程式是屬於應用於全公司的程式。在程度上來說，也可以說是為使用者量身打造的軟體，因為使用者為全體員工，此類系統為上班出勤系統，請假加班系統，文管系統等。而如果各部門或單一部門若有新增系統的須求，是否可以請 MIS 部門規劃並設計符合使用者須求的程式呢？答案是並不一定，MIS 部門因為各部門的須求不同，所以其日常工作即為各部門解決問題，解決 ERP 系統的問題，包含程式故障的排除，系統功能的增加和改變，因應客戶的須求增加功能等，以上種種在大公司，因客戶數多，產品種類多，新增的客戶多，及客戶的須求變動頻繁，所以 MIS 部門的工作並非只有軟體的新增設計，在系統的維護上也須要耗費大量的人力和資源。各部門的需求眾多，在 MIS 人力資源不夠的情形，若不是公司 ERP 系統，其他的須求，往往一件需求須耗時數月或數年，尤其各部門的須求並非全公司的須需求，並不會

得到上級老闆的重視，所以執行的效率不佳，MIS 部門並不願意為各部門的需求增加其負擔。各部門的須求，除非是公司發展的重點或是客戶稽核的缺失，否則並不會得到 MIS 部門的同意，設計符合須求的程式。

本次研究的目的，為實際學習網路程式的設計與應用，並自行設計一網路程式，為使用者量身打造，非經由購買軟體的方式，或者經由 MIS 部門設計的方式，達到需求的目的，而是透過使用者自身的學習，來設計出符合需求的程式。由於網路程式設計軟體的發達，所以本次研究為打破傳統，程式設計一定要由 MIS 部門來設計的思維，軟體的設計並非由 MIS 部門來設計，而是由使用者透過學習的方式，自行設計。

自行設計網路應用程式，對於非資訊科系學生是一大挑戰，即使是資訊科系學生也並非易事，所以本研究的目的為實際了解是否可行，並自行開發一網路應用程式系統，從無到有的實際案例學習。本研究的案例應用在某上市半導體封裝公司的維修部門，實際將維修紀錄由紙本填寫改為由電腦輸入，節省紙本浪費，改善紙本填寫的問題，不但節省紙本填寫的時間，更增加紀錄的可用性和其價值。利用網路技術，網路無遠弗屆的優點，管理者不一定要在現場才能管理現場的工作。透過網路，了解現場的維修情形，不必透過電話就可以掌控最新的維修狀況也是本研究的目的，不只解決了紙本紀錄的問題，也同時節省了資訊傳達的時間和金錢。

1.4 本論文組織

本論文共分五個章節，第一章為緒論探討本研究之背景與動機，目的與範圍。第二章文獻回顧與探討，針對現今網路程式，系統開發分析的方法，做文獻探討。第三章研究方法，以系統開發生命週期模式，做為研究方法，應用於實際案例。第四章實證分析為實際案例之網頁頁面說明，成功導入之解說。第五章結論與建議，以本次研究所得結果和問題，提供後續研究之方向。

1.5 本章結論

本研究為依研究方法將實際案例逐步完成之案例，以現今的技術來看，將表單、紙本紀錄 E 化將可改善問題和節省不少作業時間。E 化所帶來的效率改善已是不爭的事實，但要如何在最節省成本的條件下 E 化，設計出適合且實用的程式，是一個值得研究的議題。本研究為實際案例，研究和學習 VB.NET (曹祖聖，2008) 以及 ASP.NET (陳錦輝，2009) 網路程式設計的方法，應用於職場上以某上市半導體封裝廠維修部門為案例，成功導入和應用。

第 2 章 文獻回顧與探討

2.1 介紹

由於本次研究專注在應用網際網路，網頁程式的設計和應用，藉以設計和建構網路程式來改善紙本紀錄和管理的問題，達到增加管理的績效。所以本章將探討網路程式的技術，資訊系統設計研發發展和演變，系統分析方法的探討和應用等，藉以文獻探的方式，對於網路程式的應用描繪出一輪廓，並以此探討技術，以應用於實際問題的改善。

2.2 資訊系統設計研發的發展和演變

以計算機為主的資訊系統分析與設計，開始於 1950 年。1950 年以前，計算機是笨重的機器且不實用。1950 年以後由於計算機能力的提升和快速的改變，加上組織的需求，於是研發環境有了明顯的改變。在 1950 年，研發的目的，最主要是在改善軟體的效能，因為當時計算機是龐大的機器，昂貴，耗電而且不穩定。所以研發的目的，主要在於如何讓機算機的執行軟體穩定，作業有效率，節省資源。在 1950 年代，並沒有軟體工業，即沒有軟體設計公司，當時作業的軟體都是用機械語言寫的，研發最主要的是如何增加機器的記憶，機器能儲存的記憶體是很少的，所以增加機器的記憶，計算機才有用。

1960 年開始，由於技術的提升，計算機的體積開始變小，小電腦開始產生，同時價格也開始下降，此時開始有了軟體工業。此時軟體的研發和改善主要還是在改善機器效能和穩定。到了 1970 年，投入研發和工作的人力愈來愈多，此階段的研發已經有點像工程，同時也比較可以訓練，有系統化。此階段資料庫系統採用的是階層式的和網路模組，資料可存取。此時研發開始由系統優先轉向資料優先。

1980 年開始，計算機小型化，即我們現在所稱的個人電腦開始產生。個人電腦也開始成為企業最主要使用的工具。軟體設計工業開始發展的愈來愈大，更多的人投入研發。此時的軟體的語言，稱為第四世代語言，在 1970 年以前稱為第三世代，進入 1980 年以後稱為第四世代。第四代語言已不像程序式語，會告訴電腦做什麼而不是如何做。由於作業系統軟體的進步，由原來的一行一行的畫面，變成一個視窗一個視窗的應用，此時的作業系統仍為 DOS 系統，而企業所使用的應用程式，已朝圖形化開始邁進，企業所使用的軟體，由原來自行研發轉為向軟體公司購買，軟體設計研發人員的工作，由於系統的演變，逐漸的由系統的製造建立者轉為系統的整合者。

到了 1990 年末，作業系統進步到視窗系統，而應用程式研發也朝向系統整合。此時研發者使用視覺化程式設計軟體進行研發，並設計使用者的介面可以使用在客戶端和伺服器端的平台。資料庫的應用由階層化轉為物件導向的應用，公司企業所使用的資料庫系統轉為向軟體公司購置，例如 Oracle，Microsoft，Ingres 等。企業開始採購整個公司都可以應用的寬領域超大系統，不僅是應用在單一部門，而是整個企業，此類的軟體公司為 SAP AG 或 PeopleSoft。

從 1990 中開始，一直到今天為網路時代，此時研發設計的焦點為網路程式的應用，企業所使用的企業網路以及網際網路。各大軟體公司紛紛投入研發容易使用的研發程式。網路系統的執行和設計，主要有 3 個部份，首先為資料庫的部份使用一個 Server，其次應用程式為一個 Server，最後是客戶端個人電腦。2000 年開始，還有另外一個研發重點，那就是無線網路的應用。無線網路的範圍除了電腦外，2G 或 3G 手機上網已經成為趨勢 (Hoffer, et al. 2008)。

2.3 網際網路應用組成

關於網際網路的應用是由 3 個部份如圖 2-1，資料庫伺服器(Database server，本研究使用 SQL server)，網頁伺服器 (Web server)，以及客戶端 (Clients) 組成。其關係如圖 2-3 所示，互動式網頁應用程式須透過網頁伺服器編譯才可以變成瀏覽的網頁。當網路程式設計者，將寫好的互動式網頁程式存放在網頁伺服器 Web server 後，Web server 會將程式轉成 html，使用者 (客戶端 Client) 使用瀏覽器 (Explorer) 便可以瀏覽網頁和執行程式到資料庫伺服器存取資料。

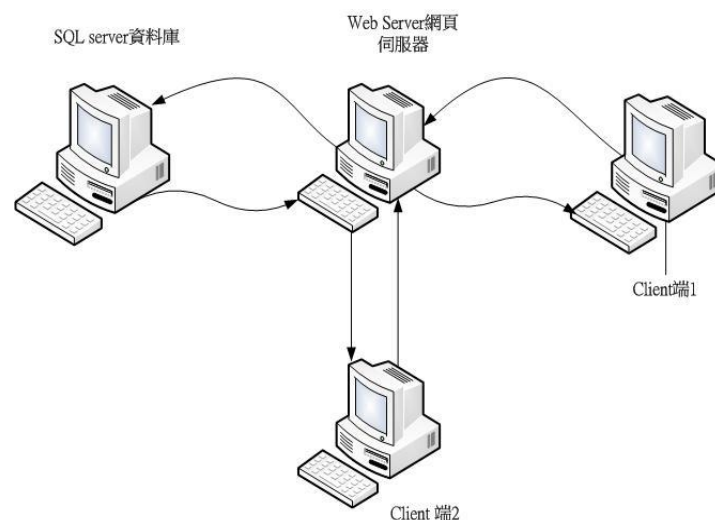


圖 2-1 網際網路應用組成圖

2.4 系統分析的方法

全球化的經濟將帶來更劇烈的競爭以及不斷的變化，如何提升生產力，生產高品質的產品及服務，維持客戶的忠誠度，資訊科技(IT)將扮演著重要的角色，甚至於是公司成敗的關鍵所在 (林國平，2007)。系統分析應用在 IT 產業由來已久，程式設計者須了解系統分析的各種方法以設計出符合須求者所須要的應用程式。透過系統分析的方法，設計者更能和須求者雙向溝通，以實際了解問題所在，才不會設計出不符合須求者或沒達到須求者滿意的程式。透過系統分析的方法，程式設計者依循系統化，邏輯化的步驟的進行，開發過程才能有系統，有效率並確保品質也更容易管理，並且減少錯誤的機會，才能開發出有用並符合須求的程式。

2.4.1 系統開發模式的演進

系統開發模式之發展大概起源於 1950 年代，如表 2-1 所示，當時最早之模式稱為編碼與修正模式，後來 Benington (1956) 提出階段模式，接著 Royce (1970) 提出瀑布模式，Mills (1971) 提出漸增模式，Bally et al. (1977) 提出雛形模式，Mills et al. (1986) 及 Boehm (1988) 提出螺旋模式，Aoyama (1993) 提出同步模式，又 Jacobson et al. (1998) 年提出 Rational 統一流程模式，這些系統開發模式，前兩者幾乎已沒人使用，其餘六種目前常被使用(吳仁和，2009)。

表 2-1 系統開發模式演進表

人名	模式	時間
N/A	編碼與修正模式	1950
Benington	階段模式	1956
Royce	瀑布模式	1970
Mills	漸增模式	1971
Bally	雛形模式	1977
Boehm	螺旋模式	1988
Aoyama	同步模式	1993
Jacobson	RUP	1998

2.4.2 瀑布模式

瀑布模式(Waterfall model) 由如圖 2-2 由 Royce (1970) 提出，針對當時的系統開發提出改善，並發表其想法為瀑布模式。Royce 打破之前，須要分層的階段模式，改為沒有固定階段的模式，視系統開發須求可分成最少兩階段，也可分成 3 階段，7

階段等。並且前後階段可以往返。瀑布模式最早強調系統開發應有完整之週期，故亦稱為瀑布式系統發展生命週期 (System Development Life Cycle, SDLC)，由於瀑布模式強調系統開發過程需有完整的規劃，分析，設計，測試及文件等管理與控制，因此能有效的確保系統品質，它已成為業界大多數軟體開發之標準 (Boehm,1988)。

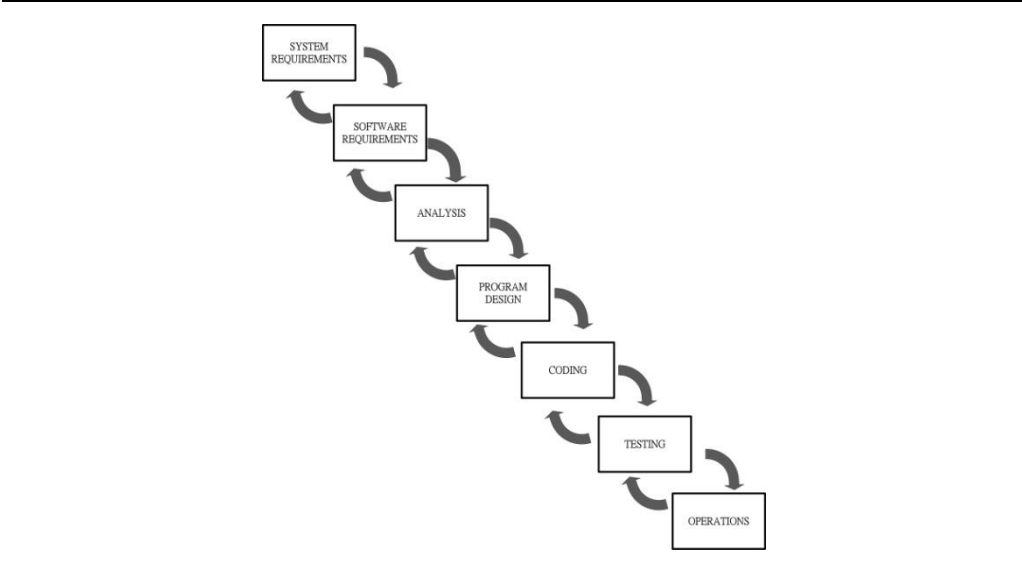


圖 2-2 瀑布模式

另外關於系統分析與設計，國內外的研究著作，在 1970 年至今如雨後春筍般，不斷有不同的應用和不同的階段定義。針對系統發展生命週期，我們約略探討如下的文獻，此大部份的文獻為國外學者之著作，國內學者採用和翻譯。

表 2-2 七階段瀑布模式發展週期之步驟

階段	內容	概念
1 問題定義	確認問題，定義原因，勾勒出解決策略	Why
2 分析	決定什麼是解決問題所必須要做的事情	What
3 設計	決定問題如何被解決	How
4 開發	建立系統。程式撰寫，硬體安裝，撰寫操作使用程序等	Action
5 測試	測試系統	
6 實施	將系統遞交給使用者	
7 維護	維持系統正常運作	

資料來源：姚明河和黃明官 (1997)

如表 2-2 為常見的階段式瀑布模式之定義，其各階段之名稱略有不同，此為 Davis W.S.於 1983 年所著 (姚銀河和黃明河譯，2003)。其階段共有七段。另外 Eliason A. L. (林淑芬譯，1992)，將系統生命週期定義為系統分析 (System Analysis)、系統設計 (System Design)、系統發展 (System Development)、系統實施 (System Implementation)、系統評鑑 (System evaluation) 等五個階段。以及 Shelly, G. B.定義五個階段，分別為初步調查 (Preliminary Investigating)、系統分析 (System Analysis)、系統設計 (System design)、系統開發 (System development)、系統實施與評估 (System Implementation and Evaluation)，不同著作之各階段名稱如表 2-3。

表 2-3 系統生命週期階段名稱比較表

作者	階段	階段名稱
Davis W.S	7	問題定義、分析、設計、開發。測試、實施、維護
Eliason A. L.	5	系統分析、系統設計、系統發展、系統實施、系統評鑑
Shelly, G. B.	5	初步調查、系統分析、系統設計、系統開發、系統實施與評估

由表 2-3 可以知不同的作者所定義的階段名稱，但其中必定有分析，設計和開發以及實施，所以階段數增減，但其內容大致相同。另外 SDLC 為反覆開發之過程，當軟體不符使用時，將開發另一軟體取代，如圖 2-3。針對以上分析，本次研究對於 SDLC 將採四個階段，於第三章中分別為系統規劃，分析，設計和執行。

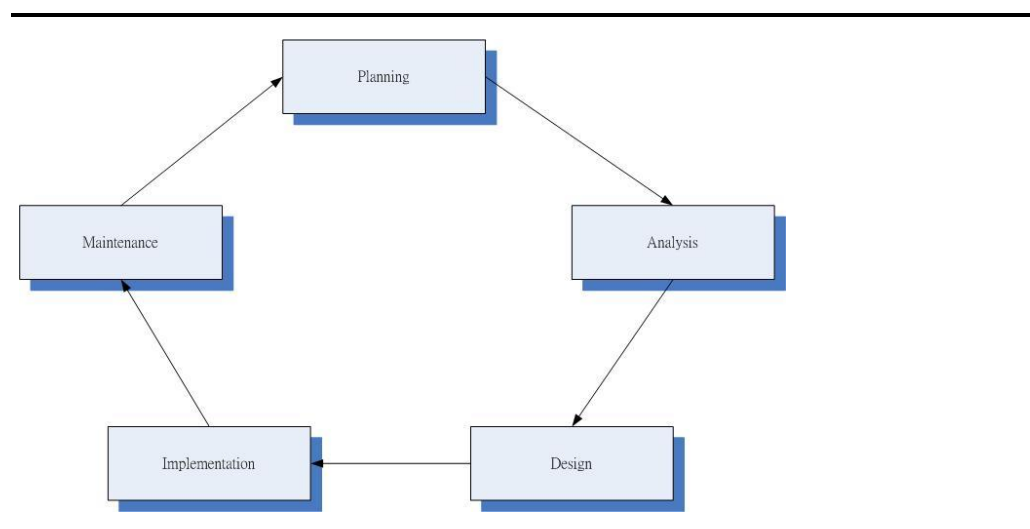
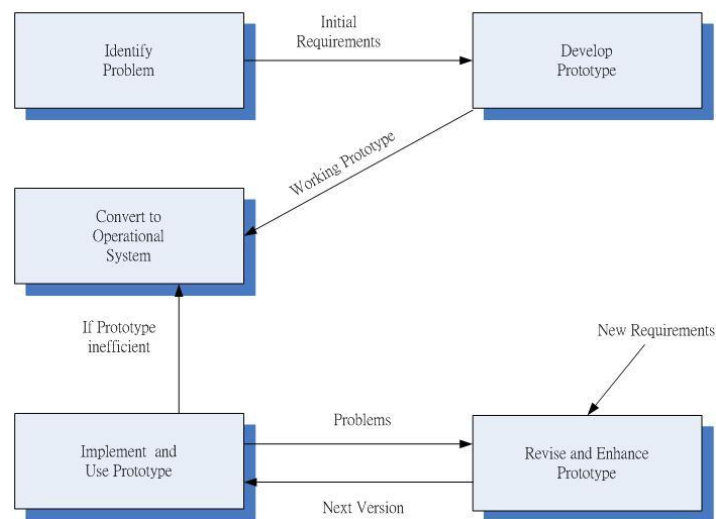


圖 2-3 系統開發生命週期 資料來源：Hoffer, et al. (2008)

2.4.3 雛形模式

關於雛形模式 (Prototyping Model)，由 Ballye et al. (1977) 提出，其針對使用者之須求已經完全了解，但還無法有方法解決，所以先開發雛形，並透過雛形實際操作，邊作邊修改，反覆檢討，直到符合須求者的須求為止，如圖 2-8。雛形模式是須求者高度參與的模式，藉由雙方（須求者和開發人員）不斷的溝通，雛形的操作與回饋，釐清，了解，修改，增加功能，到最後符合實際需求並沒有問題為止。雛形模式適用於需求改變可能發生於整個專案生命期間，客戶能高度參與，應用領域不熟悉或高度風險等情況的專案 (Brumbaugh,1991)。在本次研究中，系統開發人員亦為操作人員之一，並非純為開發人員，所以實務上雛形系統，具有快速開發，適時修正，符合須求的優點，適用於小型系統和本次研究。

圖 2-4 雛形模式



資料來源：Hoffer, et al. (2008)

2.5 本章結論

我們得知網頁程式可以利用個人 PC，另外資料庫和應用程式可同時應用於個人 PC。透過文獻探討我們也得知，系統開發生命週期模式可以依所須要的階段，從開始，分析，設計，加上執行即可依此有系統的完成開發。網頁程式的應用，方法和功能不斷的提升，我們應當用心學習並善加利用，我們應當不斷的去研究，並創造更有利的程式工具用於管理的工作，以收事半功倍之效。

第 3 章 研究方法及個案應用

3.1 方法介紹

本次研究所採用的方法為以系統分析的方法建構網路程式，亦即以上一章所探討的系統開發之分析與設計模式，所以本章將以逐步的方式實際應用和分析。本次研究，程式撰寫人員並非專職程式工作人員而是生產作業，機器設備維護的管理者，所以針對在系統生命週期中，分析者的角色和性質以及分析者的主要工作直接跳過不予以探討，直接從系統開發生命週期開始，並以我們所須要的階段，從開始，分析，設計，執行以實務案例做分析及應用。

3.2 系統開發生命週期模式

系統開發生命週期如圖 3-1 所示，本次採用四個階段由系統規劃，系統分析，系統設計，和系統執行，分別應用，其說明如以下章節。

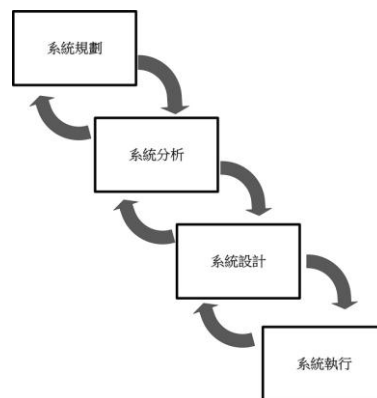


圖 3-1 系統發展生命週期

3.2.1 系統規畫

須求為第一步驟，在本次研究中，所遭遇的問題為，設備維護的管理還停留在紙本紀錄。為改善紙本紀錄的缺點，將紙本改為以系統來取代，改善工作效率。故須要建構一個網路系統，名為簽修系統其須求要點分析如下，此即為系統規劃書。

使用前的作業環境：使用者為生產線機器的操作人員，即 Operator，我們俗稱 OP 和維修人員，俗稱 RM。在還未有系統以前，當機器設備故障，OP 須要在機故登記表 (附錄 A1) 中登記，即俗稱叫修。RM 在看到機故登記表有人叫修之後，

系統的須求：當有系統以後，可以將以上的動作改為在電腦上輸入，並且不要有太多的動作，所有資料是用點選的，由系統自動帶出，不必用鍵盤輸入，減少打字的時間此為系統建立的目標。

[illegible]

圖 3-3 機器維護表之實際圖片

須求目標：系統的功能，在維修的系統建立完成之後，除了將紙本 E 化外，我們希望還要有管理的功能。例如維修人員在維修室或辦公室可以遠端監控，即可以像看網頁一樣，瀏覽機台的叫修紀錄，網頁只要有新增，畫面就會更新。另外系統須另設有維修歷史查詢功能，可供維修，生產主管查詢，並且當客戶有抱怨時，可供 QC 品管人員查詢，其使用案例圖如圖 3-4 所示。

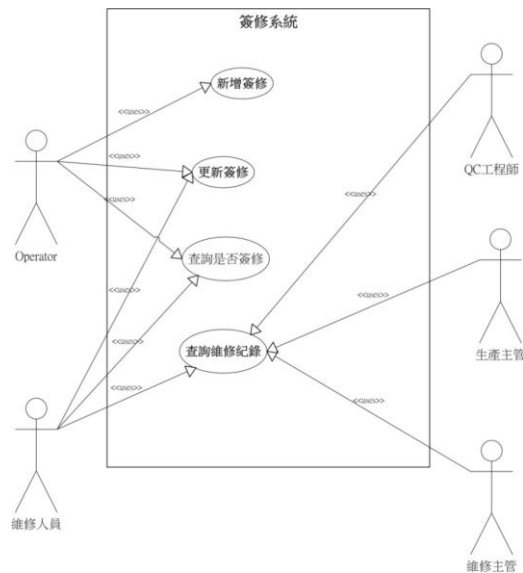


圖 3-4 簽修系統之使用案例圖

3.2.2 系統分析

簽修系統為新建立的系統，在這之前，並無其他系統，所以資料的來源重新開始，過去的紙本依然保存，當系統建立後即不再填寫而改由系統輸入，如下所述為系統須求書，系統的功能畫面須求如下。

主頁面：為登入系統之後，所出現的畫面，畫面的內容即為機故登記表，其時間排序為最近的在最上面，依序每增加一筆即排在最上面，其他的往後退，頁面須要分頁，每頁為 15 筆，每間隔一行須要以不同的顏色來顯現，以方便瀏覽，易於閱讀。

OP 簽修頁面：主要供 OP 使用，內容全部為點選，不用鍵盤輸入。

維修記錄頁面：此頁面專為修護人員所使用，維修人員點選主畫面的任何一列中的故障情形欄，即會連到此頁面並只有該筆資料，不會有別筆，以利維修人員針對該筆做處理。

故障情形頁面：供維修人新增和修改機台的故障情形資訊。此故障情資訊主要提供 OP 點選用。維修人員可以依機台種類分別輸入機台可能發生的故障情形，若遇

有未發生過的故障情形，此頁面功能必須可以隨時新增

管理者頁面：此頁面的主要功能為可以編輯所有欄位資料，此頁面為管理者頁面，簽修系統為維護資料的正確性和完整性，一旦簽修成立之後，資料就不能再做修改，但為防止輸入錯誤例如選錯機台又未及時發現取消，所以管理者須要有修正的功能。此管理者頁面並未開放給所有人員，否則資料可以任意修改將造成無公信力和資料不正確和不完整。

3.2.3 系統設計

首先系統須要建立新的資料庫，我們將以 Microsoft SQL2008 Express 為本次的資料庫軟體。簽修系統所需要的資料庫表單有以下幾項所述及如圖 3-6 為資料庫表單之關聯圖。

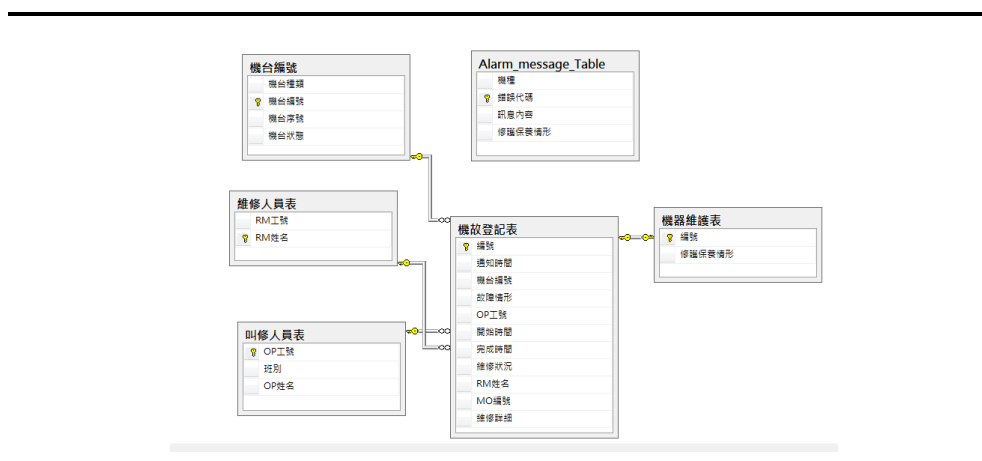


圖 3-5 須求之資料庫表單關聯圖

機故登記表：此表單將作為系統的主畫面(Default)，所有的系統頁面從主畫開始延伸和連結，主畫面為最主要的瀏覽畫面。

機台編號：記錄所有的機台種類和每一機台種類的所有機台編號。

叫修人員表：記錄所有生產線作業人員 Operator 的工號和姓名和班別。**維修人員表：**記錄所有維修人員的工號和姓名。

維修狀況表：設備的維護狀態，有待處理，維修中，已完成、待工程，保養中，調機中等狀態。

錯誤訊息表：機故發生時，OP 往往描述不清楚，或者在叫修輸入資料時，會隨意輸入，又或者打字的問題會成為 OP 的困擾，所以針對不同的機台種類，不同的故障情形，所顯示的錯誤訊息列表，讓 OP 在叫修時直接點選故障情形，將可減少叫修輸入的時間和保持資料正確性。

本次研究所使用的設備在硬體方面，只使用一台個人 PC (其規格如下表) 當主機，而連線之電腦，佈滿生產線，超過 20 台電腦同時連線，作業員在其工作區域，機器設備附近即可使用。目前此套系統已成功的運轉，並將推廣至其他部門，已獲得國外部門預約，將於 8 月份前往安裝。

表 3-1 硬體規格表

廠牌	中央處理器	記憶體	硬碟
ASUS	Intel Celeron E 1500 2.2G	1G	250G

在軟體方面，所使用的開發程式皆從微軟網站下載免費之開發軟體，其種類如下所述：

表 3-2 使用軟體表

軟體	用途	性質
Microsoft Visual Web Developer 2008 Express	ASP.NET 互動式網頁程式開發軟體 (陳錦輝，2009)	開發工具
Microsoft SQL server 2008 Express	資料庫管理工具	開發工具
Microsoft .NET Framework 3.5	網頁編譯	開發工具
Windows XP SP3(內建 IIS 6.0)	Web 伺服器	伺服器

3.3 本章結論

沒有做規劃和分析如何知道須求是什麼，所以透過 SDLC (System Developer Life Cycle) 模式，從規劃開始以條列式列出想要的功能作為目標，定出其範圍。接著做系統分析，資料是什麼，需要什麼樣的網頁頁面，有幾頁網頁，每一頁網頁的功能又是什麼。然後設計和逐步的編碼 (Coding)，連結資料庫叫出和寫入資料，最後完成系統設計並上線執行。系統上線之後，若有新的功能須求和要求改善，應該都是一小部份的改變，不會動到整個網頁的架構，可能是喜好的更改等，依雛形模式最後穩定至不做任何更改為止。

第 4 章 實證分析

4.1 介紹

本章將接續前一章 SDLC 的方法之後，以實際案例繼續延伸做網頁執行畫面的詳盡解說。此網頁程式為互動式網頁，我們以個人 PC 在實際網頁架站之後，實際實行，目前此系統已有超過 6 千筆的資料，系統已非常穩定，使用雛形模式新增須求或更改喜好的階段也已完成，系統不再做任何變動。本研究主要目的為探討學習，將網頁和瀏覽器當成應用軟體的使用介面，讓程式可以經由伺服器編譯，以網頁的畫面呈現給更多人使用 (周棟祥 et al. 2009)，所以在本章將得到實際的驗證。

4.2 程式流程

如圖 4-1，我們將依此流程圖，設計出每一步驟之網頁，並在本節中，以步驟 STEP1~STEP6 方式解說每一網頁之功能，其內容如下一頁。

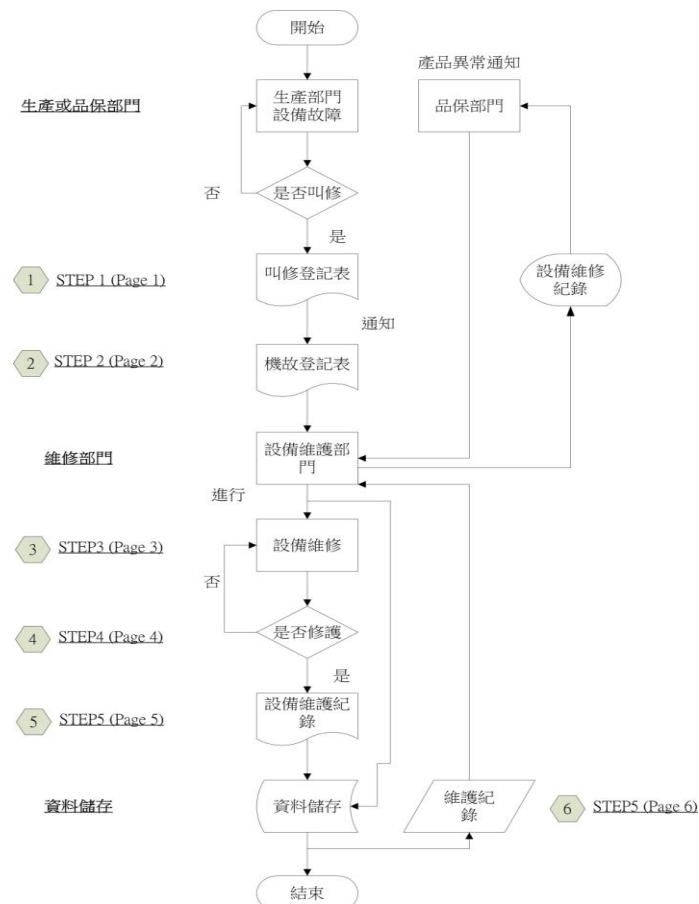


圖 4-1 功能設計流程圖

4.3 程式內容

Step1 設備故障叫修頁面：此程式頁面為生產部門，遇設備故障，須進行簽修的動作，新增一筆設備故障的登記。其頁面設計如下：

編號	通知時間	機台編號	故障情形	OP工號	開始時間	完成時間	維修狀況	RM姓名	MO編號
202	2010/4/26 下午 12:12:00	IDT001	Mark 不良(AT28)	1318			待處理	待處理	

機台種類: AT28 機台編號: IDT001

故障情形: Mark 不良(AT28)

請選擇班別: 1 OP工號: 1318

維修狀況: 待處理

RM姓名: 待處理

MO編號:

新增

圖 4-2 設備故障叫修畫面

如圖 4-2，畫面內容設計為讓使用者即作業員，輸入訊息時，全部用點選的，包含機台種類、機台編號、班別、工號、產品的資料 MO 編號，輸入完最後再按新增，即立刻新增一筆叫修紀錄，並顯示在畫面的最上方，如此叫修作業即完成，此為系統作業第一個頁面。

Step2 簽修系統主頁面：此程式頁面為簽修系統之主畫面，頁面主要設計以行列式列出所有設備之故障清單及修復情形，如圖 4-3。

請記編號 Key PROMIS Please remember to key PROMIS

機台登記表

REGISTRATION MACHINE SHUTDOWN

SPEC NO.00-R-0029

[\[OP叫修/Call Repairing\]](#) [\[Revise/更改資料\]](#) [\[RM維修/RM Repairing\]](#) [\[即時畫面/LIVE\]](#)

編號	通知時間	機台編號	故障情形	OP工號	開始時間	完成時間	維修狀況	RM姓名	MO編號	維修時間
3408	2010/5/18 15:13	IDJ015	當機(MV)	T407	5/18 15:17	5/18 15:26	完成	林重佑	1993428.1	13
3407	2010/5/18 14:55	PAB014	調機(set up)	2843	5/18 14:57		維修中	許孟喜	1997661.1	36
3406	2010/5/18 13:16	PAB017	SET UP(TR48)	YD67	5/18 13:16	5/18 13:43	完成	許孟喜	1993970.1	27
3405	2010/5/18 12:50	IDJ015	軌道卡貨	T407	5/18 12:53	5/18 13:31	完成	王重謬	1993428.1	41
3404	2010/5/18 12:43	IDT006	REJ下貨不良	YD68	5/18 12:44	5/18 13:26	完成	林重佑	1995901.1	43
3403	2010/5/18 12:42	IDW002	調機(換產品)	N646	5/18 12:43	5/18 12:53	完成	王重謬	1995689.1	11
3402	2010/5/18 12:06	IDJ015	軌道卡貨	1318	5/18 12:07	5/18 12:23	完成	蔡漢宗	1993428.1	17
3401	2010/5/18 11:55	IDW001	調機(換產品)	N646	5/18 11:55	5/18 12:08	完成	林重佑	1993428.1	13

圖 4-3 簽修系統主畫面

此頁面設計之主要欄位內容為：通知時間，機台編號，故障情形，OP 工號，

開始時間，完成時間，維修狀況，RM 姓名(維修人員)，及 MO 編號。此頁面之內容主要由作業員於機台故障時輸入，系統會自動帶出登記時間。此頁面畫面設計以條列式方式列出所有的維修紀錄，以顏色管理，維修未完成正在維修中以藍色顯示，尚未有人維修以紅色顯示，所有完成維修的紀錄，則無底色顯示。條列式畫面的好處為所有進行的活動一目了然，再加上顏色管理，可以讓維修人員或管理者很容易掌握現場的狀況並立即前往處理，此為系統作業第二個頁面。

Step3 維修處理頁面：此程式頁面為簽修系統之維修頁面，主要由維修人員進行維修活動時維修紀錄之編輯。



圖 4-4 維修編輯畫面

此頁面由 Step2 頁面點選任一列之故障情形即會連結至此頁面。此頁面列出該故障情形所屬之機台。如圖 4-4，當維修人員點選左側修復按鈕，點選的那一列會進入編輯模式，如圖 4-5 並出現起修和完修兩個按鈕，欄位內容為空白。



圖 4-5 維修開始

維修人員點選起修按鈕，開始時間欄位之內容會立即顯示現在的時間，即起修時間，且維修狀態欄之內容會自動改為維修中。此頁面之主要設計為簡化維修人員輸入的動作，時間自動抓取只要按一個按鍵並由下拉式選單點選姓名，起修之紀錄即完成。當維修人員維修並完成後，同樣的點選修復按鈕，點選完修，完成時間自動帶出，維修狀況欄自動改為完成，接著點選確定之後，維修之紀錄即完成，如圖 4-6。

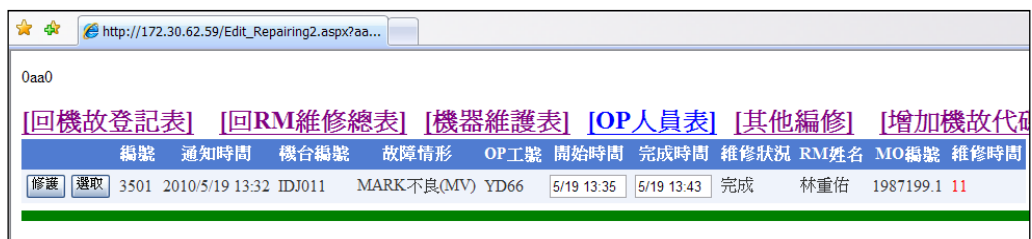


圖 4-6 維修完成圖

另外除了時間輸入的點選之外，維修人員點選左側選取按鈕即會帶出維修內容欄位畫面，如圖 4-7。此時維修人員可以按編輯按鈕進入編輯畫面並編輯內容，編輯完按確定即會存檔，此為維修之詳細內容。



圖 4-7 維修內容欄位畫面

Step4 維修未完成頁面：此程式頁面為簽修系統是否完成之頁面，若有未完成則全部列出。



圖 4-8 維修是否完成之頁面

此頁面如圖 4-8 所示，會列出所有待處理和維修中之機台，此頁面主要的作用為整理未完成維修之機台並依時間先後全部列出，不會有疏漏或未維修完成之機台。若無待維修之機台則如圖 4-9 無為設備故障之畫面。

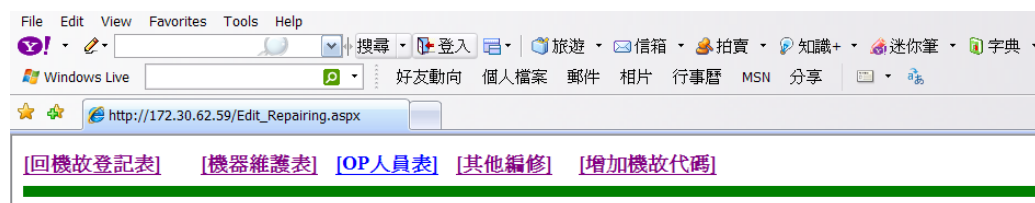


圖 4-9 無設備故障畫面

Step5 機器維護表頁面: 此程式頁面為機器維護表，其欄位和主畫面 page2 的差別在於比 page2 多一個維修詳細欄位，如圖 4-10。

http://172.30.62.59/machine_repair_sheet...

Page
Tools

機器維護表

MACHINE REPAIR SHEET

SPEC NO.00-R-0029

部門DEPT167

[回機故登記表]

[回RM維修]

	編號	機台編號	開始時間	MO編號	故障情形	維修詳細	完成時間	RM姓名
編輯	選取	3495	IDJ012	2010/5/19 10:01	1998396.1	定位不良(放不正)	調整 TAPE INDEX MOTOR 拉力	2010/5/19 10:27 賴琮建
編輯	選取	3494	IDJ019	2010/5/19 09:05	1996425.3	軌道卡貨	HST卡住排除	2010/5/19 09:07 王重諤
編輯	選取	3493	IDT008	2010/5/19 08:41	1998363.3	當機 Hang up 及工程借機	重開機及TEACH SORTER 位置,工程借機	2010/5/19 09:07 蔡漢宗
編輯	選取	3492	IDQ022	2010/5/19 08:28	1998736.1	RVSI SET UP	CHECK PNP OK	2010/5/19 08:32 賴琮建
編輯	選取	3491	IDT001	2010/5/19 08:24	1997949.3	調機(AT28) SET-UP	2Q-PS-24L	2010/5/19 08:58 許孟喜
編輯	選取	3490	IDJ012	2010/5/19 08:19	1998396.1	定位不良(放不正)	調整MOTOR STEP	2010/5/19 08:23 蔡漢宗
編輯	選取	3489	IDJ006	2010/5/19 08:19	1994986.1	定位不良(放不正)	調整INDEX補償值	2010/5/19 08:28 林重佑

圖 4-10 機器維護表

如上圖 4-10，此頁面最主要的設計為承襲紙本機器維護表的內容 (如附錄 A2) 而來。紙本的內容是以機台名稱來分類，不同機台放置於不同之資料夾，填寫或尋找時，依編號拿取，完成後再放回，如此即可以分類方便填寫和尋找，如圖 4-11。

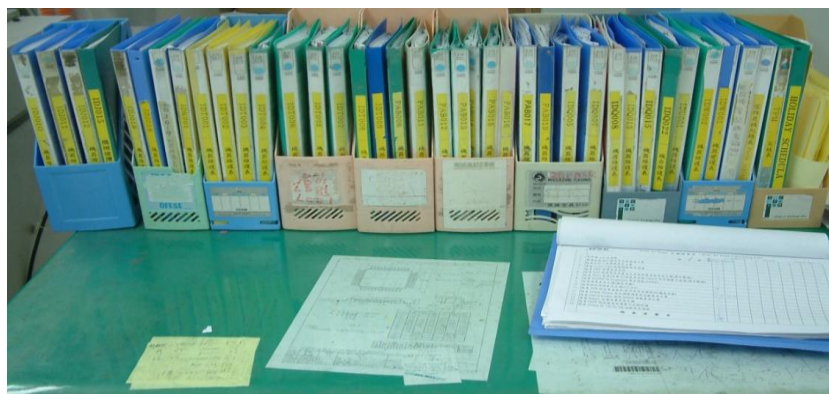


圖 4-11 以機台編號分類資料夾

機器維護表如同病人之病歷表，為機器之病歷表，或稱為維修歷史紀錄，如

圖 4-11，一台機台一本，若有設備 50 台則有 50 本，此為傳統之方式。當系統 E 化後，在系統裏我們仍保留此頁面，紀錄開始時間，故障情形，維修內容，完成時間和處理者，此頁面為簽修系統第五個頁面，除了此頁面之外，我們另外建立一搜尋頁面，依機台編號找出所有該機台的維修紀錄，如圖 4-12。

編號	日期	機台編號	RM姓名	通知時間	開始時間	完成時間	故障情形	維修分鐘	維修詳細	維修狀況
3491	2010/5/19	IDT001	許孟章	08:23	08:24	08:58	調機(AT28) SET-UP	35	2Q-P5-24L	完成
3462	2010/5/19	IDT001	Richard	00:37	00:39	00:51	調機/S(Set up for tube to tube)	14	Align the output tube	完成
3450	2010/5/18	IDT001	Richard	23:33	23:34	23:44	上料不順	11	Adjust the tube shuttle backward position and align the output tube	完成
3441	2010/5/18	IDT001	吳格昇	22:10	22:13	22:31	機台沒檔案(NO FILE)	21	SET-UP 2Y-P5-16L-TUBE	完成
3431	2010/5/18	IDT001	吳格昇	20:10	20:14	20:47	X046	37	排除卡貨, Teach Sorter 位置	完成
3428	2010/5/18	IDT001	Richard	19:42	19:45	20:02	X040	20	Align the sorter position	完成
3416	2010/5/18	IDT001	Richard	17:20	17:21	18:19	漏測(un-inspected lead)	59	shiny lead, machine cannot inspect for this product. To inform process engineer and suggest removing product to another machine.	完成
3335	2010/5/17	IDT001	吳格昇	20:06	20:10	20:25	Mark 不良(AT28)	19	Teach 3組 Mark	完成
3326	2010/5/17	IDT001	Richard	19:35	19:38	19:55	X040	20	Align the sorter stopper and teach the position	完成
3325	2010/5/17	IDT001	林垂佑	19:06	19:06	19:25	X040	19	排除卡貨	完成

圖 4-12 依機台編號列出維修紀錄

STEP 6 資料搜尋頁面:此頁面可以依不同的類別來搜尋，如圖 4-13 可以單一搜尋或做多重搜尋。若是單一搜尋，系統設定只能單選一種類別，例如圖 4-13 使用批號來搜尋，於 MO 欄位輸入批號，按確定即可完成搜尋，頁面會列出該批號所有機台的改機或維修資料，此為第 6 個頁面，品管人員或管理人員，因產品異常時可以調出相關的機台維修資料以確認產品異常原因是否為設備故障造成。

編號	通知時間	機台編號	故障情形	開始時間	完成時間	維修詳細	RM姓名	MO編號	維修狀況	維修
5795	2010/6/10 23:41	PAB016	Input Tray跳貨(TR-48)	23:48	6/10 23:53	以空 TRAY 盤讀 RUN 無卡 TRAY 的情形, KEY 回 11	黃盟舜	1996589.2	完成	12
5769	2010/6/10 18:02	PAB016	調機(set up)	18:10	6/10 18:10	Set-up vision and pick and place	高宗毅	1996589.2	完成	8
5768	2010/6/10 18:01	PAB015	調機(set up)	18:02	6/10 18:09	Set-up vision and Pick and Place	Albert	1996589.2	完成	8
5756	2010/6/10 16:36	PAB016	調機(set up)	16:39	6/10 17:28	MICREL TQFP48L7x7 拉力多次產品仍在烤箱 未SETUP p.&p	唐凱龍	1996589.2	完成	52
5755	2010/6/10 16:35	PAB015	調機(set up)	16:38	6/10 17:49		陳忻信	1996589.2	完成	74

圖 4-13 依類別搜尋

4.4 本章結論

經過 6 個步驟以實際頁面解說，可以了解網頁可以像應用程式般的操作，並沒有不同。此系統不須要像應用程式一樣，在使用時須安裝，只要連結上網路即可以進入系統操作。此為網路的方便和優勢，利用現代化技術來管理，可以提升管理的績效和為公司帶來競爭的優勢。

第 5 章 結論與建議

5.1 管理者的迷思

在工廠的工作經驗中，經常要填寫一些作業紀錄，而這些作業紀錄通常會耗費不少的填寫時間。由於工作忙碌，作業員通常為了操作機器設備或者處理產品的流程而忘了填寫。或者由於工作忙碌須要連貫的工作而無法停下來，會有等一下再寫的想法，但往往會無法停下來，以致於紀錄未填寫或不完整。同樣的道理，機器設備的維護人員，因機器設備數量多，不停的走動在各機器間，以確保設備的產出，關於維修紀錄，便會有等一下再一起寫的想法而造成忘記填寫或填寫不完整。不管是作業人員也好，或者維修人員也好，都會遇到作業填寫的問題。事實上作業紀錄的填寫會打斷作業的連貫性，所以操作設備的人通常都是在設備已作業一段時間之後，才開始補齊作業前的檢查紀錄，而維修人員也會於設備維修告一段落之後才會補齊維修紀錄。作業紀錄未寫，寫錯或者不確實的問題，存在於工廠從未間斷過。每當問題發生時，大家都有很好的意見，想要將紙本 E 化，並勾勒出一美好的願景，希望有一個系統，功能強，簡單又方便，不用費很多力氣就可以完成很多工作。但真正要提到設計或者提出須求時，卻發現大家都勸想要執行的人別傻了，因為本身不是程式設計師也不是老闆，既不會寫程式也無法有資源請人家做。所以到最後，問題沒有減少或解決過，每次被稽核都被發現一大堆紀錄缺點，此問題從未間斷過但大家也都是持續的作，並沒有任何進步。

身為管理者，在管理的過程中並無法了解和掌控每一件事，所以通常會要求下屬在下班後，每天傳送當班或當日的工作日報表。管理者只要看到日報表的內容就可以了解當天機台維修或工作的狀況。由於網路的發達，管理者都很喜歡看 mail，什麼事都用 mail 傳，所以會要求下屬，把當天設備維修的情形，用 excel 當報表，傳送給相關的人員，然後相關人員再 review 日報表的內容。看報表，Review 報表的內容，是管理者每天必做的事，但是對於下屬來說，可能就是增加其工作負擔。當下屬的人每天為應付老闆的須求，填一大堆報表，發一大堆 mail，而老闆也不見得會很認真逐筆逐字的看，因為老闆很忙，除非有異常才會仔細看報表的內容。所以日子久了，日報表已經變成一種形式，每班都要傳送，但是不是有用那就不一定了。紀錄的問題，作業紀錄再加上日報表一下子，變成同樣的紀錄要寫三次，以維修紀錄來說，機故登記表寫一次，維護表寫一次，下班後再打成報表又一次，總共變成三次。紀錄的問題，已是基層的負擔，人員填寫一堆作業紀錄，不但不能落實的填寫，並且會花費很多時間，這已突顯出管理上的爭議。主管無法了解下屬的困難，把人力資源和時間花在報表上，下屬並且延時下班，可是管理者認為這是下屬應該做的，可以延時下班，把工作完成。報表或紀錄的問題已經變成管理者的迷思，如何改善效率已不是管理者的重點，如何把事情完

成才是重點。在工廠的工作經驗中，不只是作業紀錄或者報表的問題，類似的經驗常會遇到。管理者沒有辦法或利用工具，來改善管理或作業的效率，只能用增加工作的方法來完成工作。結果員工的負擔增加，對員工的要求愈來愈多，員工要做的事愈來愈多，要填寫的紀錄愈來愈多，要打的報告愈來愈多，工作煩忙似乎是一種假相，並不是真的業務有增加或產能有增加而是工作增加，例如當產品異常被客訴時，管理者能夠用的辦法，就是增加作業的檢查，而有檢查就必須要有紀錄，才能証明有檢查，所以一個客訴就增加一個填寫紀錄，下一次不同的客訴，就必須想出不同的解決辦法，所以又增加另外一條檢查項目，同時又增加另外一項填寫紀錄。忙碌的管理者已經沒有改善效率的想法，只有完成工作的想法，如何把事情完成才是重點，所以這就是管理者的迷思，不得不增加作業的檢查，不得不增加作業規範，於是焉作業規格愈來愈厚，人員也愈來愈容易會犯錯，因為要注意易要檢查的東西太多了，但沒有很好的方法來協助。

5.2 研究結論

本次研究利用互動式網頁技術，實際設計出互動式網頁稱為簽修系統，把網頁變成是一個應用程式，網頁和瀏覽器當成應用軟體的使用介面 (周棟祥 et al. 2009)，把機器設備叫修和維修的紀錄 E 化，創造出有效率的工作環境。自簽修系統實際上線以來，被使用者所接受和喜愛，這也就是用來協助解決管理者迷思的方法。透過簽修系統，維修人員下班後再也不用打報告，只要將系統內容以報表輸出即可。以前每班人員，每人須花費 20~30 分鐘打報告，現在有了簽修系統，就可以把這時間節省下來。有了簽修系統以後，平常麻煩的紀錄問題也解決了，不會有漏寫或忘記未填的問題，每一筆記錄都是完整和可查詢的。當有產品異常，須要追查維修資料，只要輸入批號，即可完整做查詢，調出所有該批號的設備資料，且在 1 分鐘內即可完成。由於簽修系統的方便，在簽修系統上線以後，紙本紀錄即走入歷史，簽修系統是用資訊科技的方法，把繁雜的工作，簡便化，只要按一兩個鍵即可自動化完成時間的紀錄，所以節省了大量紙本的紀錄時間，使維修人員有更多的時間投入設備的維護保養，同時管理者也有更多的時間，花在問題的處理上。管理者藉由網路的便利，不論是在何處，只要有網路的地方便可以做遠端監控，透過畫面即可以知道，某台機器設備是否維修時間過久是否有困難，不需要支援，並進而到現場進行了解和協助問題處理。簽修系統成功的改善了管理的問題，增加管理的績效，節省人力資源，創造非凡的價值，簡單，方便，節省時間，對公司對個人貢獻良多，同時更重要的，系統有查詢的功能，讓管理者能對資料作分析，甚至做到資料採礦的功能，這也是未來努力的方向。透過簽修系統，可以分析，那些是有參考價值的維修資料，可以收集或查詢變成一種機器 troubleshooting 的資料庫。

附錄

附錄 A1 機故登記表

機 故 登 記 表

Registration For Machine Shutdown

部門別 Dept:

[illegible]

附錄 A2 機器維護表

機器維護表

MACHINE REPAIR SHEET

部門 DEPT :

機器/模具編號 M/C NO.: _____

年 YEAR

[illegible]

附錄 B1 簽修系統 STEP2 網頁之 code behind 程式

```
Protected Sub GridView1_RowDataBound(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.GridViewRowEventArgs) Handles GridView1.RowDataBound
    If e.Row.RowIndex = -1 Then Return '在編輯模式,則不作用
    If e.Row.RowIndex <> GridView1.EditIndex AndAlso CType(e.Row.FindControl("Label4"),
Label).Text.Trim.IndexOf("待處理") > -1 Then
        e.Row.BackColor = Color.Red '有待處理的欄位,那一列底色為紅色
        e.Row.ForeColor = Color.Black
    End If
    If e.Row.RowIndex <> GridView1.EditIndex AndAlso CType(e.Row.FindControl("Label4"),
Label).Text.Trim.IndexOf("維修中") > -1 Then
        e.Row.BackColor = Color.Orange '有處理中的欄位,那一列底色為橙色
        e.Row.ForeColor = Color.White
    End If

    '底下為超連結
    Dim hk As HyperLink = e.Row.Cells(3).Controls(0) 'cell(3)即故障情形列為超連結
    hk.NavigateUrl = "Edit_Repairing2.aspx?aa=" + DataBinder.Eval(e.Row.DataItem, "編號").ToString.Trim
    '底下為每一列最後面增加一欄並計算時間維修時間,完成減通知
    If DBNull.Value.Equals(e.Row.DataItem("完成時間")) Then '如果完成時間欄位沒有值,則用現在時間
        去減通知時間
        e.Row.Cells(GridView1.Columns.Count - 1).Text = DateDiff(DateInterval.Minute,
DataBinder.Eval(e.Row.DataItem, "通知時間"), Now())
        e.Row.Cells(GridView1.Columns.Count - 1).ForeColor = ColorTranslator.FromHtml("#FF0000")
    Else '完成時間減通知時間
        e.Row.Cells(GridView1.Columns.Count - 1).Text = DateDiff(DateInterval.Minute,
DataBinder.Eval(e.Row.DataItem, "通知時間"), DataBinder.Eval(e.Row.DataItem, "完成時間"))
        e.Row.Cells(GridView1.Columns.Count - 1).ForeColor = ColorTranslator.FromHtml("#FF0000")
    End If
End Sub
```

附錄 B2 簽修系統 STEP 3 網頁之 code behind 程式

```
Protected Sub GridView1_RowCommand(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.GridViewCommandEventArgs) Handles GridView1.RowCommand
    If e.CommandName = "startR" Then
        CType(GridView1.Rows(GridView1.EditIndex).FindControl("textbox1"), TextBox).Text =
FormatDateTime(Now, DateFormat.ShortTime)
        CType(GridView1.Rows(GridView1.EditIndex).FindControl("dropdownlist1"),
DropDownList).SelectedIndex = 1
    End If
End Sub
```

```

End If
If e.CommandName = "FinishR" Then
    CType(GridView1.Rows(GridView1.EditIndex).FindControl("textbox2"), TextBox).Text =
FormatDateTime(Now, DateFormat.ShortTime)
    CType(GridView1.Rows(GridView1.EditIndex).FindControl("dropdownlist1"),
DropDownList).SelectedIndex = 4
End If
End Sub
Protected Sub GridView1_RowDataBound(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.GridViewRowEventArgs) Handles GridView1.RowDataBound
    If e.Row.RowIndex = -1 Then Return
    If e.Row.RowIndex <> GridView1.EditIndex AndAlso CType(e.Row.FindControl("Label1"),
Label).Text.Trim.IndexOf("待處理") > -1 Then
        e.Row.BackColor = Color.Red
        e.Row.ForeColor = Color.Black
    End If
    Try
        CType(e.Row.FindControl("BUTTON1"), Button).Enabled =
(CType(e.Row.FindControl("TEXTBOX1"), TextBox).Text.Trim = "")
        CType(e.Row.FindControl("TEXTBOX1"), TextBox).Enabled =
(CType(e.Row.FindControl("TEXTBOX1"), TextBox).Text.Trim = "")
    Catch ex As Exception
        Response.Write(e.Row.RowIndex)
    End Try
End Sub

```

參考文獻

中文部份：

- 吳仁和、林信惠，(2009)，系統分析與設計－理論與實務應用，智勝文化事業有限公司。
- 周棟祥、吳進魯，(2009)，ASP.NET 專題實務，文魁資訊股份有限公司，台北市。
- 周棟祥、吳進魯，(2010)，ASP.NET 專題實務二，文魁資訊股份有限公司，台北市。
- 曹祖聖、林義証、蔡文龍，(2008)，Visual Basic2008 程式設計經典，碁峰資訊股份有限公司，台北市。
- 陳會安，(2008)，Visual Basic2008 資料庫程式設計範例教本，旗標出版股份有限公司，台北市。林國平、吳宗杉譯，2007，系統分析與設計，台灣東華書局股份有限公司，台北市。
- 陳錦輝，(2009)，ASP.NET 3.5 初學指引，博碩文化股份有限公司，台北市。
- 姚明河、黃明官，(1997)，系統分析與設計，高立圖書，台北市。
- 周坤約，(2003)，以志願模式實施體育專項選項選課之系統設計研究，中央大學資訊管理學系碩士論文。

李延平、郭鴻志，(1995)，系統分析與設計—由自動化到企業再造，華泰書局，台北市。

林淑芬，(2002)，系統開發：分析、設計與製作，基峰出版社，台北市。

陳惠貞，(2010)，網頁程式設計 HTML、JavaScript、CSS、XHTML、Ajax，基峰出版社，台北市。

英文部份：

Hoffer J. A., George J. F., and Valacich J. S., (2008), Modern Systems Analysis and Design, 5th ed., Pearson Education, Inc., NJ:New Jersey

George J. F., Batra D., Valacich J. S., and Hoffer J. A., (2008), Object-Oriented Systems Analysis and Design, 2th ed., Pearson Education, Inc., NJ:New Jersey.

Aoyama, M., (1993), Concurrent-Development Process Model, IEEE Software,. 46-55.

Bally, L., Brittan, J., and Wagner, K. H., (1977), A Prototype Approach to Information System Design and Development, Information and Management, 1(1), 21-26.

Benington, H. D., (1956), Production of Large Computer Programs, Proceeding ONR Symp., Advance Programming Methods for Digital Computers, 15-27.

Boehm, B. W., (1988), A Spiral Model of Software Development and Enhancement, IEEE Computer, 21(5), 61-72.

Brumbaugh, D. E., (1991), Object-oriented Development Building CASE Tools with C++, John Wiley and Sons, Canada.

Mills, H., (1971), Top-down Programming in Large Systems, in Ruskin, R. Ed., Debugging Techniques in Large System, Prentice Hall, Englewood Cliffs, New Jersey.

Royce, W. W., (1987), Managing the Development of Large Software Systems: Concepts and Techniques, ICSE 9, Computer Society Press.

Jacobson, I., Booch, G., and Rumbaugh, J., (1999), The Unified Software Development Process, Addison-Wesley, Reading, Massachusetts.

Davis, W. S., (1983) System Analysis and Design: A Structured Approach. Reading, MA: Addison-Wesley.

Eliaison, A. L., (1987), System Development: Analysis, Design, and Implementation, Little Brown and Co, Boston, MA.

Shelly, G. B., Cashman, T. J., Adamski, J. J., and Adamski J., (1995), System Analysis and Design, 2th ed., Course Technology, Inc., Florence, Kentucky.