物件導向軟體專案開發制度之建立與導入
——以國內鋼鐵公司之研發部門為例

黃明祥
屏東科技大學資訊管理系

摘要

物件導向開發方法的應用是近代軟體專案開發過程的一個重要研究領域，在導入一個大型物件軟體專案的過程中，需要專業化軟體開發技術及 CASE 工具的配合，但是若無建立一套完整的專案開發制度確實是一件相當困難的工作。有鑑於此，本研究根據 Rational Rose 公司發展的 Rational Unified Process（RUP）的基礎架構，整合 ISO 9000-3 的內涵與軟體工具發展出一套新的物件導向專案開發制度，提供研究者與實務界發展相關技術的參考依據。為驗證本研究提出的物件導向專案開發制度應用之可行性，以國內一家大型鋼鐵公司研發部門為實證對象，根據本研究的調查發現使用者對於建立與導入專案制度的重要需求與期望，在制度設計方面考慮的重要因素前三名分別為標準化、人員的角色與工作項目以及軟體工具的配合；至於對於專案制度內容之需求的重視程度的前三名分別為專案制度的完整性、專案制度資料內容取得的方便性與修改的彈性；其次，使用者認為影響專案制度導入成效的因素的重要性前三名分別為高階主管的重視與支持、導入適合的軟體工具、教育訓練工作；至於導入專案制度以後的主要績效前三名分別為縮短開發時程、降低開發的成本與提昇軟體品質。本制度最大的優點與特色是將制度建立在 Web 作業環境，使用者可以直接透過網路擷取相關作業標準，並予以應用在軟體專案的開發過程，因此導入與應用十分容易且學習時間較短，上述的研究成果可以提供軟體業者或資訊部門設計軟體專案制度之參考。

關鍵字：物件導向方法、軟體專案管理、電腦輔助設計工程工具

Development and Implementation of an Object-Oriented Software Project Management System: An Empirical Study in the R&D Department of a Steel Company

Ming-Shang Huang
Department of Information Management,
National Pingtung University of Science and Technology

Abstract

Recently applications of OO methods become an important research topic in software project management. Implementations of OO software projects call for the applications of sophisticated software developing techniques and powerful software tools. However, it is very difficult in managing a large-scale and complicated OO software project without the support of a comprehensive management system. Therefore, the purpose of this paper is to develop an objected oriented software project management system to solve the above problems. The proposed system is based on the principles of the Rational Unified Process (RUP) developed by Rational corporation and ISO9000-3. To examine the feasibility of the proposed system, an empirical study in the R&D department of a steel company is conducted. Key issues in developing OO software project development systems are identified by using the questionnaire. The findings of the survey show that standards, role of software project developers and support of software tools are ranked top three factors in establishing project management systems; the comprehensiveness of project management systems, ease of use, and elasticity of improvements are ranked top three factors in the applications of project management systems; management supports, selection and applications of software tools and user trainings are ranked top three in the implementing process; reductions of development times, reductions of development costs and improvements of software quality are ranked top three major benefits of implementing project management systems. The advantages of the proposed system are adaptive and easy to be implemented on the web environments. Finally, some useful guidelines for implementing software project management systems are pointed out.

Keywords: Object Oriented Methods, Software Project Management, CASE Tools.

壹、前言

近年來，軟體開發的工作環境，已經有了相當明顯的改變，例如軟體產品趨於複雜化、軟體開發涉及異質性資料庫、大型軟體的開發工作需求也逐漸增加，而且幾乎大部份的軟體必須在網路的作業環境下執行，因此現代軟體開發的工作面臨相當大的挑戰。基本上，軟體專案的開發是一種技術與腦力密集的工作，軟體開發工作大部份是由一群人共同合作完成的，對於軟體的開發工作者而言，確實有必要建立一套完整的作業制度與標準的工作環境。由於傳統軟體開發工作的重點在於技術的引進，因此「軟體專案管理」(Software Project Management) 長久以來一直不被重視，導致軟體品質不佳與生產力低落的現象 (Yourdon, 1992)。

一般而言，影響軟體開發最重要的因素是 4P (People, Project, Product, Process) (Cooper and Fisher, 1979)，其中以 "人" 的因素最為重要，軟體專案的人力資源管理確實相當重要 (Kan et al., 1994)。由於軟體開發涉及的因素相當複雜，若沒有建立一套完整的軟體專案開發制度，極易導致軟體專案失敗的結果，軟體設計的品質與生產力亦極為低落。另一方面，軟體開發工作完成以後，軟體的維護工作可能面臨相當大的困難，一旦專案人員離職，許多資訊系統立即面臨嚴重的軟體危機問題。因此，軟體開發的管理制度與作業流程是不容忽視的一環。由於大部份資訊從業人員對於軟體開發的管理制度並不了解，經常導致軟體品質與生產力低落的結果 (Abdel-Hamid, 1996; Boehm, 1981; Humphrey, 1989)。

有關國內軟體專案開發工作所面臨的問題，茲予以整理如下：(謝遠敦等，民國 86；林信惠等，民 84)：(1)專案經常延誤交期，導致軟體開發成本大幅度增加，(2)使用者對於軟體品質滿意程度並不是相當高，主要的原因是軟體不能達到使用者需要的功能，(3)未建立一套標準化與系統化的開發制度與流程，(4)軟體文件欠缺或不完整，軟體維護管理相當不易，(5)軟體業者不重視軟體標準 (例如：ISO 9000 系列標準等) ，與(6)軟體品質管理人員未徹底執行確認與驗證工作。根據以上分析，軟體專案管理制度的建立對軟體業而言是一項相當重要的工作，它也是一個大型企業邁向技術自主、技術創新與具備高科技研發能力的重要基石，因此受到大型企業資訊部門或軟體業者的高度重視。

近年來，物件導向方法逐漸成為軟體開發的一種重要開發方法。根據專家的研究報告 (Boggs and Boggs, 1999; Rumbaugh et al., 1999; Jacobson et al., 1999)，利用物件導向技術或工具進行軟體開發工作具有下列的優點：(1)建立軟體開發的標準作業環境，據以提升軟體開發工作的品質與生產力，(2)將專案人員從事於軟體開發的工作流程、經驗與技術建立一個完整的知識庫管理系統，而達到軟體再用(Reuse)的效果，(3)建立軟體開發的工作團隊 (Team Work)，全面提升軟體的設計能力，(4)利用物件導向開發工具發展的資訊系統可以在不同的作業平台執行，與(5)利用物件導向開發工具可以大幅度提升作業效率與軟體品質。有鑑於此，本研究擬針對物件導向專案之軟體專案制度的工作，進行深度的分析與探討。

以建立物件導向專案之軟體專案制度而言，下列問題是研究者重視與關心的課題：

1. 一個物件導向軟體開發制度應具備哪些特性？
2. 物件導向軟體專案開發工作應如何進行？
3. 建立專案管理作業流程應考慮哪些重要因素？
4. 建立專案管理作業制度應如何進行的步驟？
5. 一個完整的專案管理作業制度應包含哪些內容？

根據上述說明，本研究擬對於建立軟體開發制度的過程，作深入的分析與討論。茲將本研究目的說明如下：

1. 探討建立一套物件導向軟體專案開發制度之重要議題。
2. 建立一套完整的物件導向軟體專案開發制度與流程，作為軟體業者設計專案開發制度之參考。
3. 根據本研究發展的模式予以實際應用在企業研發部門的軟體開發過程，據以發現企業導入物件導向軟體專案開發制度的問題並提出一些具體建議供軟體業者之參考。

貳、文獻探討

本研究相關的文獻擬分為以下三個部分進行論述：

（一）專案管理的相關文獻

一般而言，專案管理（Project Management, PM）可以定義如下：「利用人、資源、系統與技術，成功的完成某一專案之管理程序。」（王慶富，民 85；Cleland et al., 1988；Liu and Horwitz, 1989）。換言之，專案管理的目的在於結合人員、設備、資金及時間等資源（Resources），在截止日期及預算範圍以內，完成預定之工作項目為主要目標。以軟體專案而言，可以將它分為以下幾個階段（圖 1）：(1)定義階段，(2)設計階段，(3)程式撰寫階段，(4)系統測試階段，(5)驗收階段，與(6)建置與正式作業階段（Rakos, 1990）。

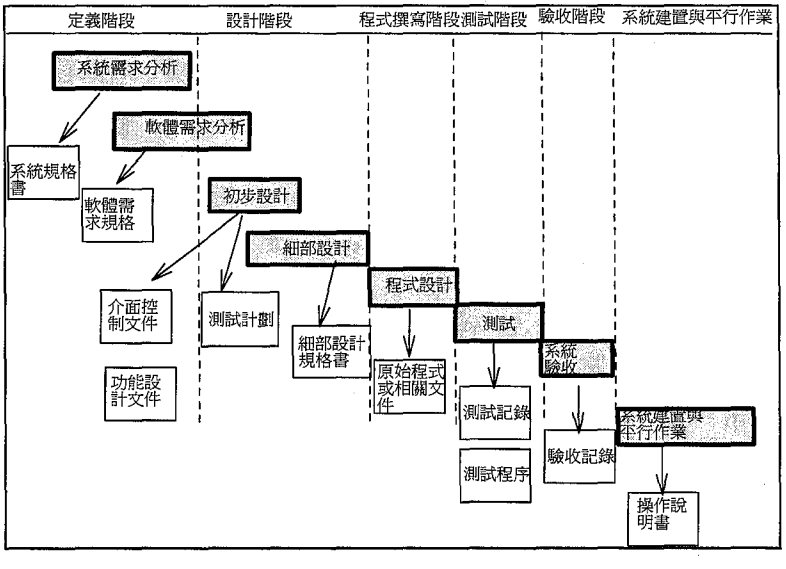


圖 1、軟體專案開發的程序

如圖一所示，軟體專案實施的第一階段就是定義問題，主要的工作為成立專案小組，進行問題分析以及擬訂專案計劃書。專案計劃書主要內容包括：簡介、執行計劃、組織計劃、測試計劃、更改控制計劃、文件計劃、訓練計劃、檢討及報告計劃、安裝及作業計劃、資源分配與交貨計劃等；『設計階段』的工作係針對系統需求的內容進行設計工作，它包括系統規格書及程式規範書的撰寫、資料庫檔案設計等工作；『程式撰寫階段』主要工作為完成軟體程式的設計與單元測試工作；『系統測試階段』是由測試小組根據系統規格進行測試工作；『驗收階段』係由客戶對專案小組完成的軟體進行驗收工作；『系統建置與平行作業階段』係將舊有系統予以轉換到新的作業環境，因此現場測試工作有其必要性，此外在系統正式作業前，必須對使用者進行操作訓練工作，同時利用在系統正式作業以前的平行作業期間，可以發現一些尚未發現的問題並予以澈底的解決。以軟體專案管理制度與流程的建立而言，主要是涵蓋以下工作項目：(1) 設計軟體開發階段與管制點，(2) 標準作業程序，(3) 設計標準文件與表格，(4) 規定參與軟體設計人員的工作執掌，與 (5) 軟體專案管理制度的導入與推行（李坤清，民 83）。為達到上述目標，下列工作的配合仍然相當重要：(1) 軟體專案管理制度的教育訓練講習，(2)先導專案計畫的執行與修正，以及(3) 建議與引進一些適合的軟體專案管理的軟體工具（Berkeley et al., 1990）。

（二）物件導向方法的相關文獻

物件導向設計方法的主要重點在於物件模式（Object Modeling）的建立與應用（Pressman, 2001）。一般而言，將複雜的資訊作業需求，經由抽象化過程可以減少系統複雜度（Complexity），同時，物件導向方法可以將資訊系統的需求分析與設計作業達到無縫隙（Seamless）之良好效果，利用物件模式、動態模式與功能模式表示資訊系統的功能，對於資訊系統的發展與軟體的品質有相當大的助益。基本上，物件導向方法具有下列優點（Bennett et al., 1999）：抽象化（Abstraction）、資訊隱藏（Information Hiding）、包含（Encapsulation）、模組化（Modularity）、繼承（Inheritance）、同質異型（Polymorphism）、訊息傳遞（Message Passing）、抽象化層次（Level of Abstraction）、與再用性等（Reuse）。因此，有關物件導向技術的研究重點在於物件相關資訊的表達（Representation）以及設計類型（Design Pattern）的建立與運用。

採用物件導向的 CASE 工具進行資訊系統開發工作，基本上是以使用個案（Use Case）作為出發點，從以下四個觀點進行系統分析與設計工作： (1) 結構觀點－表達物件內部結構或物件彼此的關係，可以利用 Class Diagrams、Object Diagrams 等表示之，(2) 行為觀點－表達物件的動態行為或物件彼此間的訊息傳遞，可以利用 Sequence Diagrams、Collaboration Diagrams、Statechart Diagrams、Activity Diagrams 等表示之，(3) 導入觀點－表達物件在導入階段方法，可以利用 Component Diagrams 等表示之，(4) 環境觀點－表達物件在實際作業的環境，可以利用 Deployment Diagrams 等表示之。另一方面，物件導向開發技術有一項重要的特色：著重於抽象化層次與重點導向的作業方式。因此在應用此一技術的過程中，必須針對系統的需求區分各種不同等級，將整體作適當

的區分，從系統、次系統、模組、到基礎模組為止。因此，物件導向的開發方法主要的優點為：(1) 利用抽象化方式將系統作適度的簡化，(2) 降低整體系統的複雜度，(3) 系統化的導入方式，與 (4) 降低系統失敗的風險。

根據物件導向 CASE 的方法論，發展一個物件導向應用系統的生命週期包括以下四個階段（表 1）：(1) 計畫階段（Inception Phase），(2) 構思階段（Elaboration Phase），(3) 建置階段（Construction Phase），與(4) 移轉階段（Transition Phase）。基本上，採用「重複發展」(Iterative Development) 的作法，不必一定先前工作完畢以後，後續作業才能開始，因為在軟體發展過程經常會遇到需求不明確的情形，如果能夠採用以上作法不僅可以降低軟體開發的風險，而且對軟體品質的提升也有相當大的助益。同時，資訊系統發展過程彼此可以互相支援，當軟體元件發展進行重複作業（Iteration）結束時，若該軟體元件已經達到可以整合的程度而且內部軟體元件經過檢驗合格時，即可進行釋放（Release）的工作。至於，如何進行重複作業的規劃則視技術方面的需求而定，它包括新技術的作業需求、使用個案（Use Case）與架構等。整體而言，採用重複與漸進式的作法可以獲致以下的具體成效：

1. 在起始期建立企業個案（Business Case）可以降低重大的風險與驗證其正確性（Proof-of-Concept）。
2. 完成細部設計階段以後各種潛在風險與不確定性程度可以大幅降低。
3. 將軟體開發成本、缺點與開發時程減少至最低程度，因此軟體開發工作必須達到特定領域之軟體再用程度。
4. 避免延遲交貨、成本劇增（Cost Overrun）、軟體品質不良的情形發生。
5. 由於採取重複的開發方式，軟體可以做適度的調整而不至於影響整個軟體交貨時程。

表 1、物件導向發展的生命週期

階段別	主要進行的活動	主要目的
計畫階段 (Inception Phase)	■ 確認風險所在 ■ 將需求以使用個案(Use Case)建立資訊的基礎架構 ■ 估計範圍、成本、時程、產品品質 ■ 建立企業個案(Business Case)	建立目標與計畫
構思階段 (Elaboration Phase)	■ 確認影響風險的關鍵因素 ■ 建立使用個案 ■ 擴展行的架構 ■ 建立專案計畫 ■ 成本估計 ■ 完成企業個案	建立基礎架構

建構階段 (Construction Phase)	■ 發展模式與軟體 ■ 進度報告 ■ 發展應用軟體	完成系統的建置工作
轉移階段 (Transition Phase)	■ 修改軟體 ■ 除錯 ■ 教育訓練 ■ 資訊系統建立	正式推出軟體產品

資料來源：本研究整理

Rational Unified Process 簡稱為 RUP，它是 Rational 公司所發展出來一套以物件導向為基礎的軟體開發作業流程。它主要是提供一個完整的軟體開發之流程架構（Process Framework），提出一些軟體發展團隊工作的建議指引，針對軟體開發階段規定必須完成的軟體產品，工作的分派、軟體開發人員的責任均作明確的定義，因此，任何一個軟體開發部門可以根據 RUP 的內涵進行修改與擴充，設計出新的作業流程。茲將 RUP 的特色整理如下：

1. 採取漸進與重複式的流程開發方法—軟體的開發以“元件”為基本單位，如圖 2 所示，它是採取漸進與重複式的開發過程，從規劃、分析與設計、導入、測試與評估、部屬等一系列過程，允許迴圈式的循環作業過程。此一作法的優點是可以有效的解決需求改變引發軟體不斷的問題，降低軟體開發風險、早期檢測出缺點，以及增進軟體的再用性。

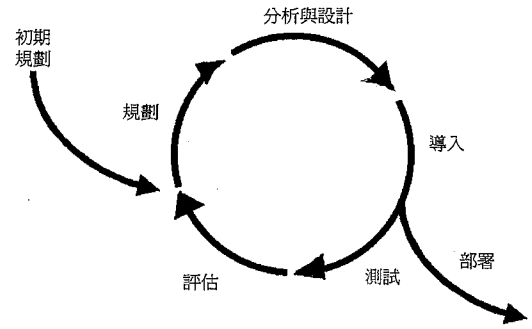


圖 2、漸進與重複式軟體開發方式

2. 充分管理使用者需求—採用系統化的方式管理使用者需求包括需求的闡述、組織、溝通與改變，它可以有效控制複雜的專案，降低軟體成本、作好群組溝通、提升軟體品質與客戶滿意度。
3. 利用元件式的基礎架構—軟體開發的初期主要是產生軟體元件它包括元件的架構類型、設計準則、限制條件，主要的優點是明確定義模組化架構、軟體再用性、允許重複式的設計。
4. 目視塑模—重視模式的建立，採用 UML 塑模，可以充分表達軟體的需求與功能。
5. 不斷驗證軟體品質—對於程序與產品進行嚴格的管控。

6. 控制軟體的更改—結合形態管理的方法對於軟體的更改作好追蹤與管理。
7. 整合各種軟體工具—將多種軟體工具予以整合並導入軟體開發流程之中。

基本上，RUP 係採取「使用個案驅動」(Use-Case-Driven) 的作法，因此，使用個案被用來定義一個系統的功能與需求。如圖 3 所示，RUP 的核心流程包括：企業塑模、建立需求、分析與設計、導入與測試等幾個階段，在每一個核心流程均採用對應的模式予以描述，例如在需求分析階段，是採用使用個案描述軟體需求，在分析與設計階段是採用設計模式（類別圖、循序圖、活動圖.....等）。

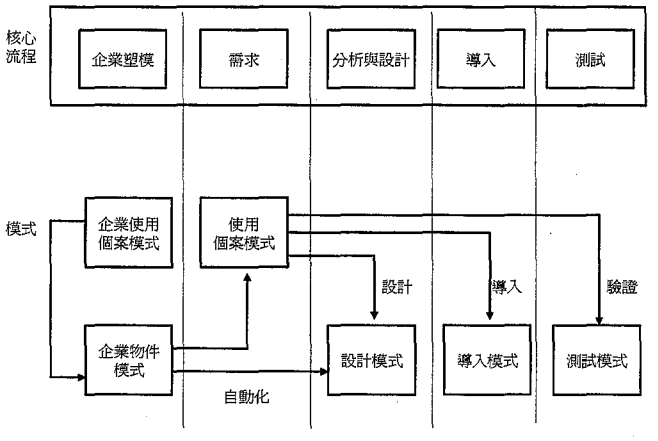


圖 3、RUP 的系統開發流程

如圖 4 所示，RUP 的軟體專案開發方法係採用「重複與漸進式作法」(Iterative and Incremental Approach)。在軟體專案進行過程中，以重複多個階段進行，每一階段均針對使用者需求進行規劃、設計與導入產生一個發行版本 (Release)，採取漸進式的修正，一直到軟體開發完成為止。同時，在軟體專案階段中，將軟體工具予以整合至軟體開發流程之中。此種作法的優點是降低專案開發工作的複雜度與軟體開發的風險，提升軟體的品質與生產力。軟體工程大師 Jacobson et al. (1999) 認為此種作法有以下優點：(1) 儘早掌握重要與顯著的風險，(2) 提出一個軟體發展的整體架構(Architecture)，(3) 擁有可以適應不可避免的修改彈性(Flexibility)，(4) 以漸進的方式發展而非一次解決的作法，整體資訊系統的開發成本較低，與 (5) 軟體設計人員擁有較高的生產力。

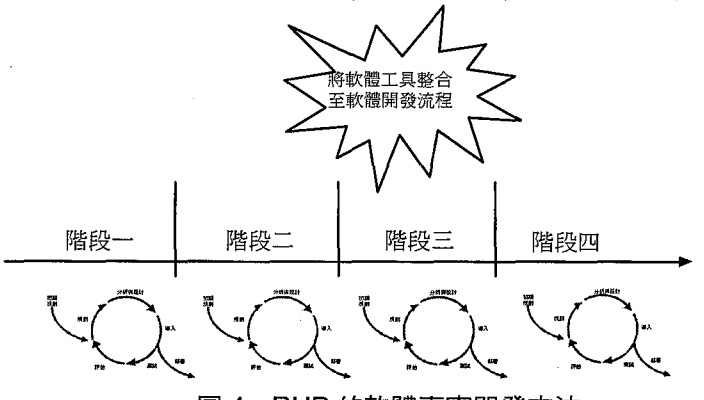


圖 4、RUP 的軟體專案開發方法

物件導向管理學會(Object Management Group, OMG) 在 2001 年提出模式驅動架構(Model-Driven Architecture, MDA)的概念，模式驅動的方法論係採用所謂「演化式的做法」(Evolutionary approach)，針對分散式異質性的資訊系統的開發需求，制定一些模式的交換標準。因此，MDA 的方法是建立在正規化或半正規化模式的基礎上，詳細的規定標準化的系統規格，而 Metadata 就是達到交換目的一個相當重要的基礎元件，任何一個符合 MDA 開發方式的資訊系統必須能夠儲存、管理與發佈應用層或系統層次的 Metadata，促使一些應用程式、工具、資料庫與其他元件均能夠加入該作業環境（圖 5）。基本上，MDA 主要的模式有平台無關的模式(Platform-independent models, PIM)與特定平台模式(Platform-specific models, PSM)兩種類型。首先，定義須 PIM，PIM 可以利用 UML 描述之，再根據 PIM 予以轉換為 PSM，而產生的 PSM 可以是 J2EE, CORBA, .NET 的作業環境。如圖一所示，為了確保共享之 Metadata 能夠達到有效的支援效果，MDA 規定任何元件必須採取下列做法：(1) 利用正規化語言表示 Metadata，(2) 制定交換格式作為發佈與交換 Metadata 用途，(3) 採用一些可以處理 Metadata 的語言工具，與 (4) 提供一些 Metadata 的服務功能等。

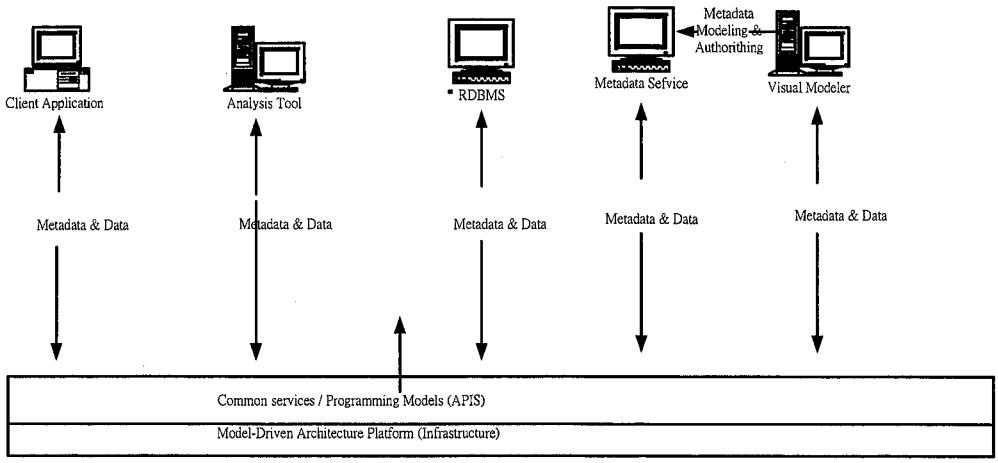


圖 5、模式驅動的方法論的應用實例

資要來源: http://www.omg.org/mda/mda_files/Model-Driven_Architecture.pdf

基本上，模式驅動的方法論之應用必須採取下列標準，其中包括：(1)Unified Model Language(UML),以物件導向學會規定的標準化圖形與語言為主，(2) Meta Object Facility(MOF),它是一種描述 Metamodel 規格的共通性與抽象化的語言，MOF 是提供一些泛用性 MOF 物件與關聯的抽象化模式，以及將 MOF 之 Metamodel 轉換至 CORBA IDL 規則，它可以定義生命週期中有關 MOF 之 Metamodel 元件的語意，(3) XML Metadata Interchange(XMI),定義 XML 當中如何以一些 XML tags 來表示 MOF-compliant 模式的做法，(4) Common Warehouse Metamodel(CWM)，主要是用來描述資料倉儲與企業分析領

域之企業與技術方面的 Metadata，其目的是針對異質性之軟體開發環境，提供不同軟體廠商開發完成之元件的交換用途，因此，MDA 方法對大型軟體專案之開發工作能夠產生極為明顯的效益。整體而言，模式驅動的方法論的優點如下：(1) 大幅降低軟體開發的成本，(2) 縮短軟體專案的開發時程，(3) 提昇軟體的再用性及軟體的品質，(4) 可以快速的融入其他新的開發技術，與 (5) 軟體具有良好的移植性、調適性與交換性。因此，模式驅動的方法論是一項適合在異質性開發環境之軟體專案的物件導向開發方法 (Ambler, 2003; France et al.; Uhl, 2003)。

（三）ISO 9000 系列標準的相關文獻

至於 ISO 9000 系列標準有關軟體開發制定的標準為 ISO 9000-3，它是針對軟體發展的過程、軟體供給與維護，規定一個最低軟體品質的標準，適合於軟體產業的應用。事實上，ISO 9000-3 亦可以視軟體產品生產者與購買者之間的一種契約。ISO 9000-3 的範疇方面，主要涵蓋以下主要活動 (Schmauch, 1995) ：(1)管理責任(Management Responsibility)：供應商、外包廠商的職責，(2)品質管理系統(The Quality Management System)：品質管理體系的文件、品質管理計畫，(3)內部品質稽核(Internal Quality System Audits)：有關於系統品質進行內部稽核工作，與(4)修正活動(Correct Action)：對於缺失進行修正活動。

其次，在軟體發展生命週期中，必須重視與執行以下的活動：(1)合約審查(Contract Reviews)，(2)採購需求(Purchasing Requirements)，(3)發展規劃(Development Planning)品質規劃(Quality Planning)，(4)設計與導入(Design and Implementation)，(5)測試與驗證(Testing and Validation)，(6)驗收程序(Acceptance Procedures)，與(7)維護過程(Maintenance Procedures) 。至於在軟體發展過程中，必要時亦需要執行以下的輔助活動(Supporting Activities)：(1)形態管理 Configuration Management)，(2)文件控制(Document Control)品質記錄(Quality Records)，(3)度量(Measurements)，(4)規則、實務與慣用語(Rules, Practices, and Conventions)，(5) 工具與技術(Tools and Techniques)，(6)採購(Purchase)，(7)整合外部的軟體(Integration of External Software)，與(8)訓練(Training)等。

參、研究流程

本研究主要目的是建立一個物件導向之專案管理制度，作為發展一些新的理論與驗證模式的主要依據，由於國內從事於此方面的研究並不多，而且進行制度的建立與深入討論的研究較少，軟體業者對於此方面的需求卻十分殷切，因此本研究的成果將有助於未來推展物件方法之應用。茲將本研究進行的步驟說明如下：

（一） 確立研究的問題—首先，從技術、流程、組織與人等不同構面探討軟體開發的重要問題，並且針對軟體專案開發工作的作業流程進行深入的分析，整理軟體專案開發之作業需求。

（二） 文獻探討—其次，蒐集軟體工程技術與專案管理之相關文獻，主要目的是歸納出

有關導入軟體開發制度與專案管理的重要議題。本研究的主要參考文獻包括主要的學術期刊、教科書、研究報告與技術手冊等。

（三） 建立研究模式—為建立一個完整的軟體專案開發制度與流程，本研究提出一個觀念性架構作為設計的依據（表 2）。如表 2 所示，第一構面為軟體專案開發階段，它包括：系統規劃、軟體設計、程式製作、軟體測試、驗收與維護等，第二構面為專案管理制度的項目，它包括：作業標準、主要工作、系統文件、系統表格、軟體工具與主要參與人員等。本研究認為一個整合性的軟體專案管理制度，應涵蓋作業標準、主要工作、系統文件、系統表格、軟體工具與主要參與人員等項目，而且應詳細規範專案開發的每一個階段的工作內容，如此一來，才能符合軟體專案開發的工作需求。

（四） 軟體專案開發制度需求分析與設計—根據個案公司之研發部門對於專案管理制度之需求進行分析，並實際進行訪談與問卷調查的工作，同時，根據物件導向開發的作業流程，設計一套適合於研發部門使用之專案管理制度，為方便與使用者使用系統文件表格，將專案開發制度的內容與作業標準、文件、表格放在 Web 上供使用者擷取與應用。

（五） 軟體專案管理工具的導入與應用—配合軟體開發階段的需求，評估與選擇適合的軟體工具，並引進至研發部門之軟體開發的過程，在引進軟體工具之過程中，進行密集的教育訓練工作以縮短引進軟體工具的時程，發揮立竿見影的成果。

（六） 新的軟體開發制度的檢討與改進—將設計完成的制度進行導入工作，每週舉行專案小組會議，參加會議成員包括專案小組負責人、參與建教合作人員、規劃與設計人員、使用者代表等。經過小組討論以後彙整改進之意見，再進行制度修正的工作。

（七） 撰寫研究報告—將最後的研究成果包括專案開發制度的流程、作業標準、文件格式、文件表格以及 Web 系統的操作手冊等資料，予以整理成研究報告。

表 2、制定軟體專案開發制度與流程之觀念性架構

軟體開發階段 / 專案管理制度的項目	系統規劃階段	軟體設計階段	程式製作階段	軟體測試階段	驗收階段	維護階段
作業標準						
主要工作						
系統文件						
系統表格						
軟體工具						
主要參與人員						

肆、實證研究 —以國內鋼鐵公司之研發部門為例

一、個案公司背景分析

本論文的實證研究對象是國內一家大型鋼鐵公司（簡稱A公司），主要產品為冷軋鋼品、線條、鋼圈、L型鋼等。根據A公司的統計資料，一年產能約為805萬公噸，其中產品的外銷比率為20%，國內外銷比率為80%。A公司的研發部門在鋼鐵專業技術研發與創新表現，獲得國際知名度相當高的評價。由於研發部門的工程師大部份均具有相當高的專業性的技術研發能力，平日累積的知識主要以軟體為主，因此如何累積研發的知識是個案公司研發部門一項重要的課題。根據研發部門的內部檢討會議，目前A公司之研發部門存在下列問題：(1) 無法有效的累積新技術開發的經驗，(2) 研發技術與知識無法共享，(3) 推動大型的研究計畫相當困難，(4) 軟體品質不高且軟體設計的生產力並不十分理想，以及 (5) 研發人員一旦是離職，對於後續的維護工作相當困難。

另一方面，A公司為因應日益競爭的國際化、自由化與新的資訊科技的衝擊，以及「知識經濟時代」來臨，擬加強軟體設計與大型軟體專案計畫的管理能力，決定由研發處成立一個專案小組，取名為『物件導向專案制度的建立』的工作，由副處長負責整個導入的工作，為了引進最新的軟體工程技術，乃與國內一家國立大學資管系簽訂建教合作計畫，著手進行軟體專案開發制度與流程的建立工作。此一建教合作的目的在於建立物件導向軟體專案管理制度、提升軟體設計的生產力與品質與建立專業知識庫。因此，透過軟體專案管理制度的建立可以達到以下的效果：(1) 建立標準的作業流程，提升軟體設計品質，(2) 引導軟體開發工作的進行，提升軟體設計生產力，與 (3) 建立完整的系統文件，累積研發的技術。

(一) 專案管理制度之建立與導入經過

本研究計畫推動期間為期一年，推動期間是從民國九十年八月起至民國九十一年七月，它主要是涵蓋以下六個階段，茲分別敘述如下：

1. 軟體專案制度之需求分析階段

此一階段主要工作為分析研發部門人員（自動化與系統檢測發展組）從事軟體開發的方法，它包括軟體開發的作業程序、軟體開發使用的表格與系統文件、軟體開發工具....等，為了達成任務乃制定一個推動的主計畫，針對每一個階段應完成的工作進度與工作內容作明確的定義，作為專案追蹤與控制的依據。在本研究計畫執行期間，每週均定期召開一次專案會議，由一位教授帶領三位研究生與專案小組人員進行工作進度的追蹤與實際問題討論，為確保本研究設計的專案制度能夠符合使用者的作業需求，在設計制度以前進行一次個案公司的研發人員的問卷的調查工作，其主要目的是充分了解使用者對軟體專案發制度建立與導入的想法，據以建立一個適合於個案公司研發部門使用之軟體專案管理制度。

基本上，本研究在問卷方面探討的重點置於以下四個構面(圖六)：(1) 使用者對專案制度的需求，(2) 設計專案制度應考慮的因素，(3) 影響推動過程的重要因素，與 (4) 導入專案開發制度的績效。如圖 6 所示，本研究調查的重點係以使用者對於軟體專案制度的需求之觀點進行分析與探討，主要目的是發現使用者對於軟體專案制度設計方面重視的因素以及對於軟體專案制度的期望，對於個案公司發展一個新制度而言，上述的資料是設計制度相當重要的資訊來源。至於問卷發放的對象是以個案公司的研發人員為主，個案公司的研發人員總人數為 120 人，本研究為使回收問卷的資料符合代表性與客觀性的需求，本研究係採用隨機抽樣的方式採集受測者的樣本，總共得到問卷十二份，茲將樣本的基本資料整理於表 3。

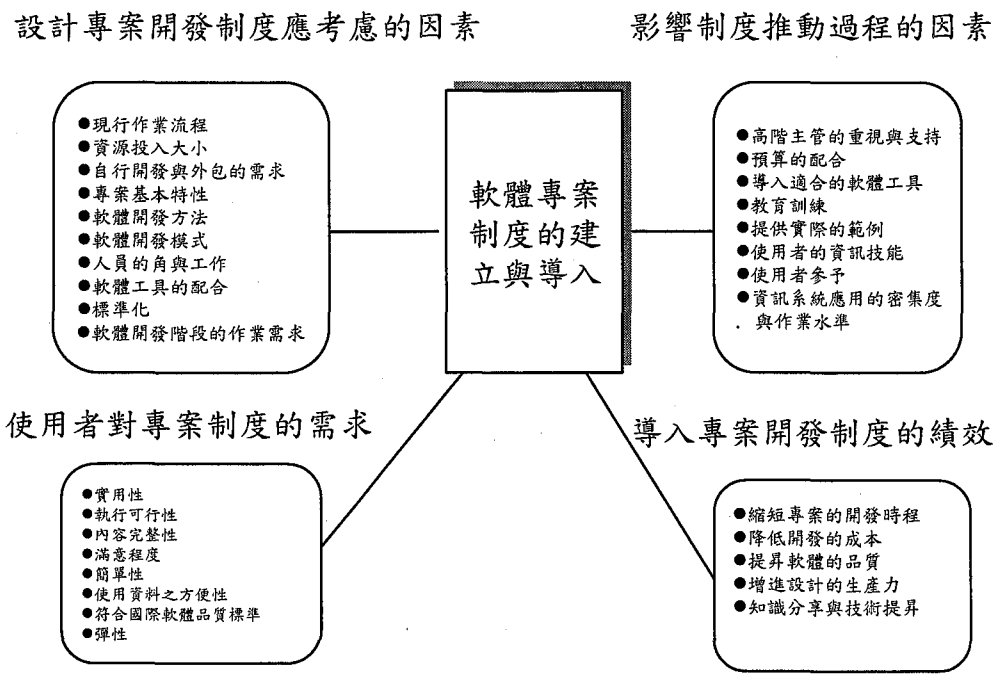


圖 6、軟體專案發制度建立與導入過程的分析架構

表 3、樣本的基本資料

基本資料	類型	人數
■ 職位	□ 主管	1
	□ 非主管	11
■ 年齡平均數	32 歲	
■ 學歷	□ 碩士	11
	□ 學士	1
■ 性別	男性	11

	女性	1
■ 工作年資平均數	5 年	
■ 研發過程中曾經扮演的角色	□ 計劃主持人或協同主持人	9
	□ 軟體開發人員	9
	□ 軟體外包管制人員	1
	□ 系統維護人員	2
■ 程式語言的設計能力 (C 、Fortran 、VB、JAVA 、C++ 、Pascal 或其他程式語言)	□ 有	12
	□ 無	0
■ 軟體開發工作的經驗	□ 有	12
	□ 無	0

茲將回收問卷的資料整理如下：

(1) 專案制度設計應考慮之主要因素

如圖 7 所示，使用者認為設計一個專案制度應考慮因素重要性程度之前三名分別為標準化、人員的角色與工作項目以及軟體工具的配合。

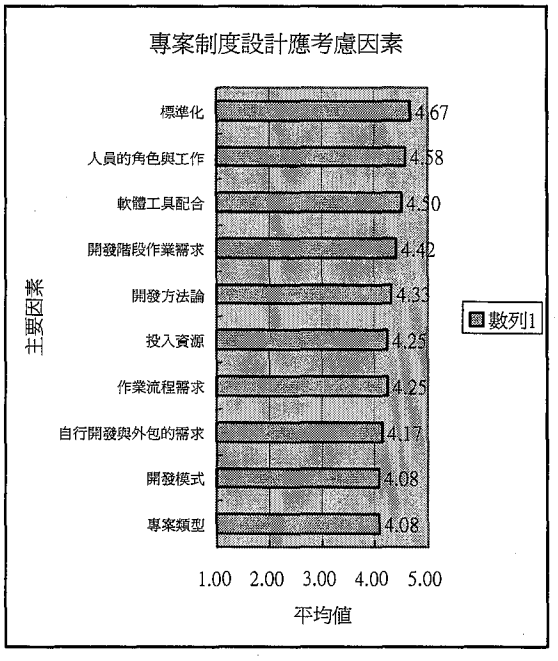


圖 7、專案制度設計應考慮的因素之重要性程度

(2) 使用者對專案制度的需求

如圖 8 所示，使用者對專案制度需求的重視程度的前三名分別為專案制度的完整性、專案制度資料內容取得的方便性與修改的彈性。

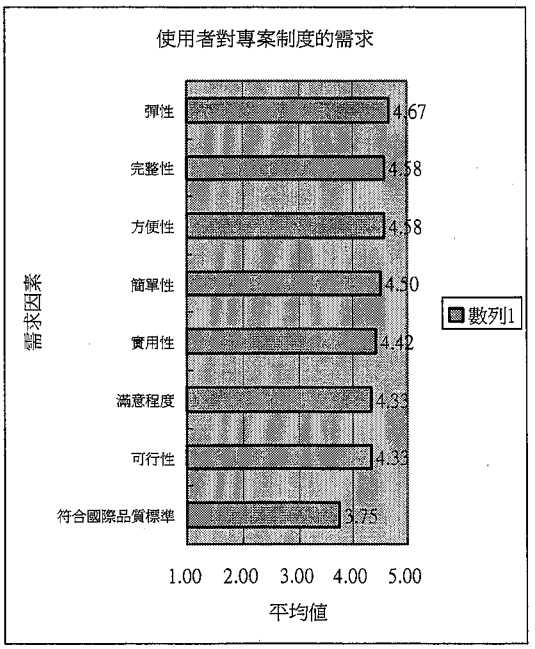


圖 8、使用者對專案制度的需求之重視程度

(3) 影響專案制度導入成效的因素

如圖 9 所示，使用者認為影響專案制度導入成效的因素的重要性前三名分別為高階主管的重視與支持、導入適合的軟體工具、教育訓練工作。

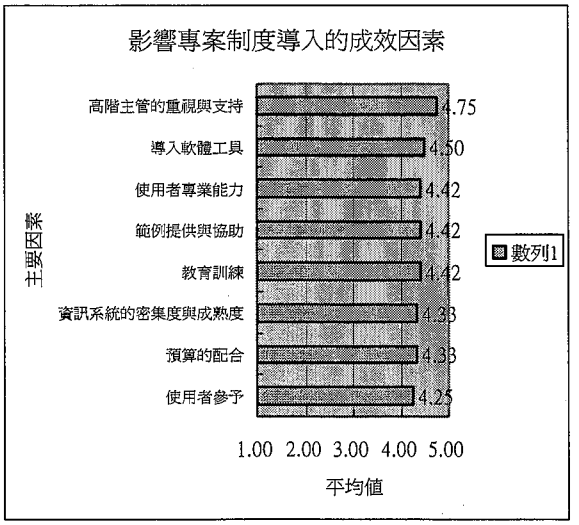


圖 9、影響專案制度導入成效的因素之重要性

(4) 導入專案制度的主要績效

如圖 10 所示，使用者認為導入專案制度以後的主要績效前三名分別為縮短開發時程、降低開發的成本與提昇軟體品質。

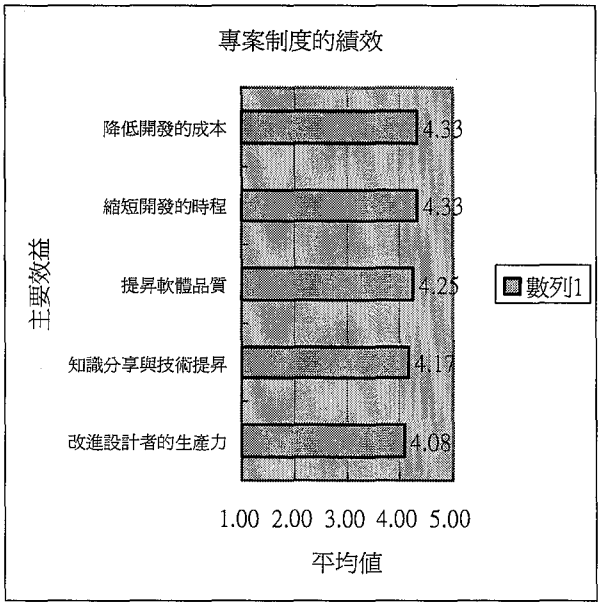


圖 10、導入專案制度的主要績效

本研究所發展的制度係根據以上使用者的需求，進行物件導向軟體專案開發制度的設計工作。

2. 作業流程規劃階段

此一階段主要工作為規劃研發部門進行專案計畫過程之軟體開發作業流程、外包軟體作業流程、.....等。軟體開發流程規劃工作主要的工作重點為結合研發工作與軟體設計工作，並評估一些適合引進的軟體工具包括專案管理軟體(Project 2000,...等)、形態管理工具(PVCS,....等)、視覺化系統分析與設計軟體(Rose, Visio, ...等)。

規劃完成的軟體專案管理制度之作業流程主要是將軟體專案之開發分為六個階段，它包括系統規劃、軟體設計、軟體制作與驗收管理、維護等六個階段。同時，針對每一階段的主要工作內容、作業標準、與系統文件均作明確的規範，軟體品質保證與形態管理之作業標準規範也包括在內。至於軟體外包專案管理制度之作業流程予以說明如下：外包管理作業涵蓋以下四個階段：軟體外包申請與核准、外包期中管制、外包軟體驗收與維護等階段。同時，針對每一階段的主要工作內容、作業標準、與系統文件均作明確的規範。

3. 軟體專案管理制度的設計階段

此一階段主要工作為根據規劃的內容建立軟體開發的標準作業程序、文件與人員

的工作職責等。此一階段主要是參考國際軟體開發標準 ISO 9000-3 架構，設計適合研發部門使用之軟體開發的標準程序與制度，並涵蓋軟體開發過程之系統開發表格。至於完成的軟體專案管理之作業標準制度內容，依據 ISO 9000-3 的規定制定詳細的條文，它包括目的、適用範圍、權責、執行工作內容、相關的表格與文件等。至於軟體外包專案管理之作業標準制度內容如下：針對軟體外包管理之各項作業標準的作業需求，依據 ISO 9000-3 的規定制定詳細的條文，它包括目的、適用範圍、權責、執行工作內容、相關的表格與文件等(詳見附錄一)。

4. 物件導向方法開發制度之設計階段

此一階段主要工作是建立「物件導向方法之軟體開發作業流程與制度」。此一階段的工作重點是根據物件導向開發方法作業的需求，規劃軟體專案之開發作業流程、系統文件與表格。物件導向方法之軟體專案管理制度之作業流程，依據物件導向軟體專案管理制度規定，整個軟體開發階段區分為六個階段，包括系統規劃、軟體設計、軟體製作、軟體測試、驗收階段、維護等階段（圖 11）。同時，針對每一階段的作業標準、主要工作項目、系統文件與系統表格等均作明確的規範。物件導向方法之軟體專案管理之作業標準制度（表 4）係針對各項作業標準的作業需求，依據 ISO 9000-3 的規定制定詳細的條文，它包括目的、適用範圍、權責、執行工作內容、相關的表格與文件等。物件導向方法之軟體外包專案管理制度之作業流程（表 5）涵蓋以下四個階段：軟體外包申請與核准、外包期中管制、外包軟體驗收與維護等階段。同時，針對每一階段的主要工作內容、作業標準、與系統文件均作明確的規範。物件導向方法之軟體外包專案管理之作業標準制度針對各項作業標準的作業需求，依據 ISO 9000-3 的規定制定詳細的條文，它包括目的、適用範圍、權責、執行工作內容、相關的表格與文件等。

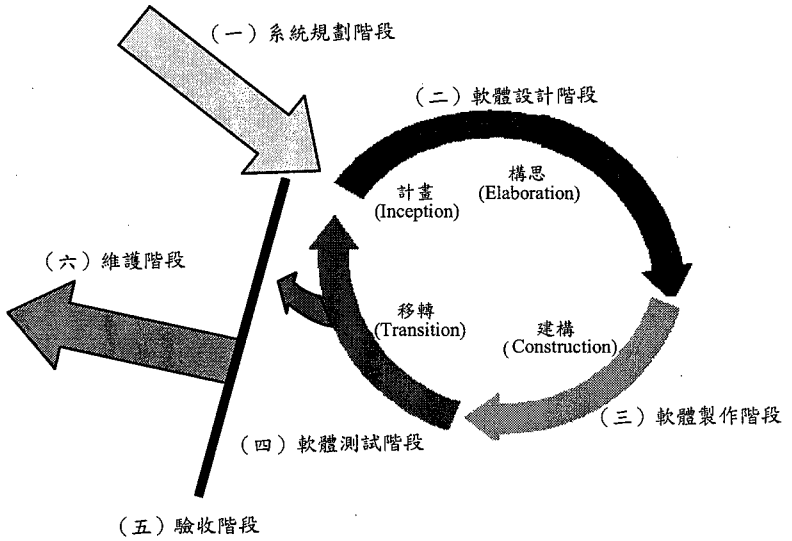


圖 11、物件導向方法之軟體專案管理作業流程

5. Web 設計階段

此一階段主要工作是建立企業內部網路(Intranet)作業環境、開發一個軟體專案管理之整合性 Web 系統，據以提供軟體開發流程所需的標準制度和系統文件管理。為使軟體標準化工作易於推展起見，規劃與建立一套在研發部門區域網路作業環境下之 Web 系統，將軟體開發過程所需的標準規範、系統文件表格、教育訓練講義與教材，均建立在 Web 系統，可以及時提供重要的專案資訊供研發工作人員之參考。圖 12 是大型專案計畫之整合性開發環境之 Web 畫面，為使專案管理制度推行順利起見，本研究建立一個整合性軟體專案管理系統，將軟體開發需求的標準制度、系統文件、作業流程予以整合在此一系統。它可以透過A公司的企業內部網路，將研發人員的系統開發文件進行集中式管理，作好定期追蹤管制，並且配合拷貝軟體的導入，建立一個完整的軟體文件管理系統。圖 13 是修正完成以後之物件導向方法軟體專案開發過程的作業系統，為使專案管理制度推行順利起見，建立一個整合性軟體專案管理系統，將軟體開發需求的標準制度、系統文件、作業流程予以整合在此一 Web 作業系統。它可以透過A公司的企業內部網路，將研發人員的系統開發文件進行集中式管理，作好定期追蹤管制，並且配合拷貝軟體的導入建立一個完整的軟體文件管理系統。

表 4、物件導向方法之軟體專案管理的制度與流程

【提案報告】 【期中報告】 【期末報告】

階段別	(一) 系統規劃階段		(二) 軟體設計階段		(三) 軟體製作階段	(四) 軟體測試階段	(五) 驗收階段		(六) 維護階段
作業標準	■ 專案規劃作業標準(T16-P10)	■ 系統規劃作業標準(T16-P11)	■ 軟體需求分析作業標準(T16-P20)	■ 軟體設計作業標準(T16-P21)	■ 程式製作作業標準(T16-P30)	■ 測試作業標準(T16-P40)	■ 系統驗收作業標準(T16-P50)	■ 系統安裝作業標準(T16-P51)	■ 軟體維護作業標準(T16-P60) ■ 系統異動與修改作業標準(T16-P61)
主要工作	■ 專案計畫之研擬 ■ 整體資源的規劃 ■ 計畫發展時程與稽核點 ■ 選擇適當的發展技術與工具 ■ 研擬軟體品質保證制度 ■ 研擬軟體型態管理制度	■ 系統需求分析 ■ 產生初步調查報告 ■ 定義使用個案 ■ 選擇硬體與軟體 ■ 定義作業程序 ■ 定義系統的界面 ■ 產生基本使用個案 ■ 審查系統規格的相關文件	■ 軟體需求功能分析 ■ 資訊流程分析 ■ 軟體界面分析 ■ 產生細部使用個案 ■ 概念性模式 ■ 測試規劃 ■ 審查軟體規格的相關文件	■ 概念性設計 ■ 細部設計 ■ 軟體界面設計 ■ 資料庫設計 ■ 審查軟體設計的相關文件	■ 程式命名 ■ 程式設計 ■ 單元測試 ■ 程式除錯 ■ 審查程式的相關文件	■ 修正細部使用個案 ■ 模組測試 ■ 整合測試 ■ 系統測試 ■ 功能測試 ■ 驗收測試 ■ 審查軟體測試的相關文件	■ 製作使用者操作手冊 ■ 製作系統人員技術手冊 ■ 系統文件驗收 ■ 軟體驗收 ■ 審查軟體驗收的相關文件	■ 系統安裝 ■ 教育訓練講習 ■ 平行作業 ■ 正式作業 ■ 審查軟體安裝的相關文件	■ 系統的維護作業 ■ 系統的異動與修改 ■ 軟體版本的更新 ■ 定期審查維護作業的相關文件 ■ 定期審查軟體異動與修改的相關文件
系統文件【註】	□ 專案計畫書（由研究計畫提案取代) (T16-F10)	□ 系統規格書 (T16-F11) ● 基本使用個案 ● 情境描述 ● 系統介面 ✧ 工作分解圖	□ 軟體需求分析規格書 (T16-F20) ● 細部使用個案 ● 基本類別圖 ● 互動圖（序列圖 或 合作圖）	□ 軟體設計規格書(T16-F21) ● 細部類別圖 ● 物件圖 ● 互動圖（序列圖 或 合作圖） ✧ 活動圖 ✧ 狀態圖 ✧ 元件圖	□ 程式規格書 (T16-F30) ● 原始碼 ● 目的檔 ● 執行檔 □ 軟體設計規格書(T16-F21) ● 佈署圖	□ 軟體測試說明書(T16-F40)	發行版本的系統文件： □ 系統規格書 (T16-F11) □ 軟體需求規格書(T16-F20) □ 軟體設計規格書(T16-F21) ● 程式規格書 (T16-F30) ● 軟體測試說明書 (T16-F40) ● 系統操作手冊(T16-F51) ● 系統技術手冊(T16-F52)	● 系統安裝手冊 (T16-F53)	□ 系統管理維護手冊 (T16-F60)
系統表格	■ 軟體發展成員指派單(T16-A10) ■ 軟體文件核准單 (T16-A11)				■ 程式清單一覽表 (T16-A30)	■ 軟體測試記錄表(T16-A40) ■ 軟體測試異常處理記錄表 (T16-A41)	■ 系統驗收記錄表(T16-A50)	■ 系統安裝與異常處理記錄表 (T16-A51)	■ 系統管理維護處理記錄表(T16-A61) ■ 軟體需求變更申請表(T16-A62) ■ 軟體變更記錄表 (T16-A63)
軟體工具	◆ Project 2000 ◆ Project Central	◆ Project 2000 ◆ Rational Rose	◆ Project 2000 ◆ Rational Rose ◆ Soda	◆ Rational Rose ◆ Soda ◆ Visio	◆ Rational Rose ◆ C++, Java, Visual Basic.....	◆ Rational Rose	◆ Rational Rose	◆ Project 2000 ◆ Project Central	◆ Project 2000 ◆ Project Central ◆ Clear Case L/T
主要參與人員	‖ 專案經理 ‖ 使用單位主管	‖ 資深系統工程師 ‖ 使用單位人員	‖ 系統分析師 ‖ 使用單位人員	‖ 系統設計師 ‖ 資料庫管理師 ‖ 使用單位人員	‖ 程式設計師 ‖ 資料庫管理師 ‖ 系統分析師	‖ 測試工程師 ‖ 使用單位人員	‖ 專案經理 ‖ 資深系統工程師 ‖ 使用單位人員	‖ 系統工程師 ‖ 使用單位人員	✧ 系統工程師 ✧ 使用單位人員

資料來源：本研究整理【註：□ 代表主要系統文件，● 代表次要系統文件，✧ 代表補充系統文件】

表 5、物件導向方法之軟體外包之專案管理作業流程

軟體外包申請與核准管制點　　軟體外包期中管制點　　軟體外包驗收管制點

階段別		(一) 軟體外包申請與核准階段		(二) 期中管制階段	(三) 軟體外包的驗收階段		(四) 軟體外包的維護階段
作業標準		■ 軟體外包申請與核准作業標準 (T16-O-S10)	■ 軟體外包廠商審查與評估作業要點 (T16-O-S11)	■ 軟體外包期中管制作業標準(T16-O-S20)	■ 軟體外包驗收作業標準 (T16-O-S30)	■ 軟體外包安裝作業標準 (T16-O-S31)	■ 軟體外包維護作業標準 (T16-O-S40) ■ 軟體外包的異動與修改作業標準(T16-O-S41)
主要工作		■ 軟體外包需求分析 ■ 軟體外包需求案件的申請 ■ 軟體外包案件的審查與核准 ■ 審查軟體外包申請的相關文件	■ 尋求合格的外包廠商 ■ 審查外包廠商的資格 ■ 評估與選擇合作的外包廠商 ■ 審查外包廠商的相關文件	■ 進度管制 ■ 軟體文件的審查 ■ 軟體品質的審查 ■ 審查軟體外包期中報告的相關文件	■ 核對交貨的軟體外包項目 ■ 驗收測試 ■ 驗收軟體外包文件 ■ 審查軟體外包驗收的相關文件	■ 系統安裝 ■ 教育訓練講習 ■ 平行作業 ■ 正式作業 ■ 審查軟體外包安裝的相關文件	■ 軟體外包的維護作業 ■ 軟體外包的異動與修改 ■ 軟體外包版本的更新 ■ 審查軟體外包異動與修改的相關文件
系統文件	整體（部份）系統外包	□ 系統規格書 (T16-F11) ● 基本使用個案 ● 劇情描述 ● 系統界面 ✧ 工作分解圖		□ 系統規格書 (T16-F11) □ 軟體需求規格書 (T16-F20) □ 軟體設計規格書 (T16-F21)	發行版本的系統文件： □ 系統規格書 (T16-F11) □ 軟體需求規格書(T16-F20) □ 軟體設計規格書(T16-F21) □ 程式規格書 (T16-F30) □ 軟體測試說明書 (T16-F40) □ 系統操作手冊(T16-F51) □ 系統技術手冊 (T16-F52)	系統安裝手冊 (T16-F53)	系統管理維護手冊 (T16-F60)
	程式外包	□ 程式規格書 (T16-F30)			□ 程式規格書 (T16-F30)		
系統表格		■ 軟體外包申請表 (T16-O-A10)	■ 軟體外包廠商審核與評估表(T16-O-A11) ■ 軟體外包廠商評估表 (T16-O-A12)	■ 軟體外包期中報告表 (T16-O-A20)	■ 軟體外包之驗收記錄表 (T16-O-A30) ■ 軟體外包程式清單一覽表 (T16-O-A31) ■ 軟體外包之驗收測試記錄表 (T16-O-A32) ■ 軟體外包之驗收測試之個案資料表(T16-O-A33)	■ 軟體外包安裝與異常處理 ■ 記錄表(T16-O-A34)	■ 系統管理維護處理記錄表 (T16-A61) ■ 軟體變更需求申請表 (T16-A622) ■ 軟體變更記錄表(T16-A63)
軟體工具		◆ Project 2000 ◆ Rational Rose	◆ Project 2000 ◆ Rational Rose	◆ Project 2000 ◆ Rational Rose	◆ Project 2000 ◆ Rational Rose	◆ Project 2000 ◆ Project Central	◆ Project 2000 ◆ Project Central ◆ Clear Case L/T
主要參與人員		I 專案經理 I 外包廠商主管	I 資深系統工程師 I 外包廠商承辦人	I 資深系統工程師 I 外包廠商承辦	I 專案經理 I 資深系統工程師 I 外包廠商承辦人	I 系統工程師 I 外包廠商承辦人	I 系統工程師 I 外包廠商承辦人

資料來源：本研究整理【註：□ 代表主要系統文件，● 代表次要系統文件，✧ 代表補充系統文件】

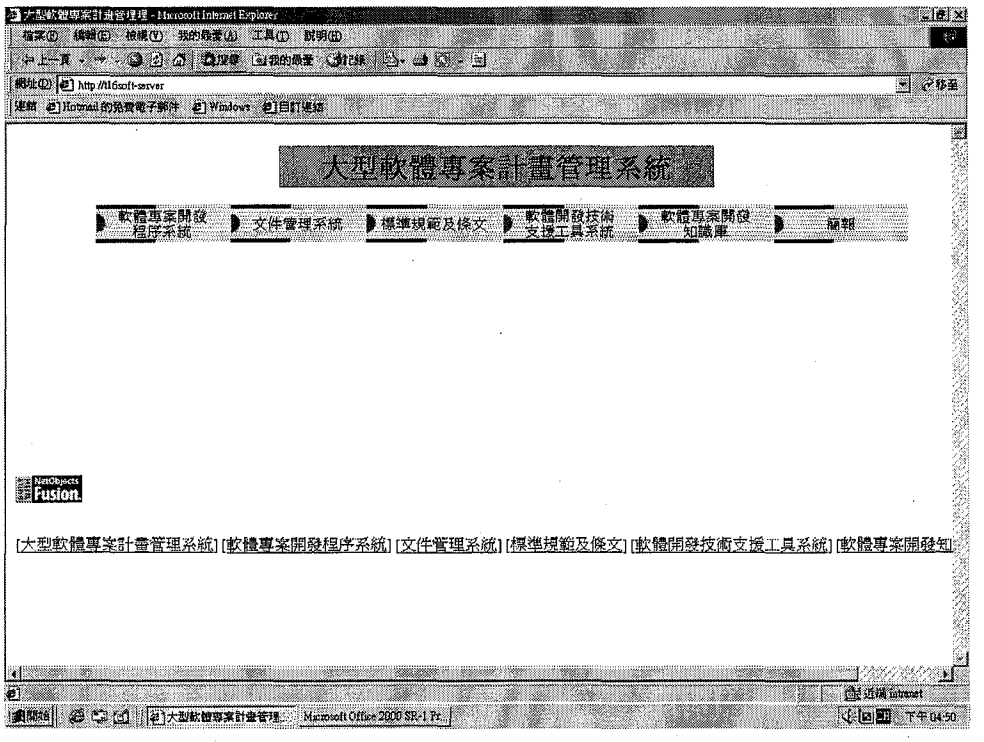


圖 12、大型專案計畫管理系統之 Web 畫面

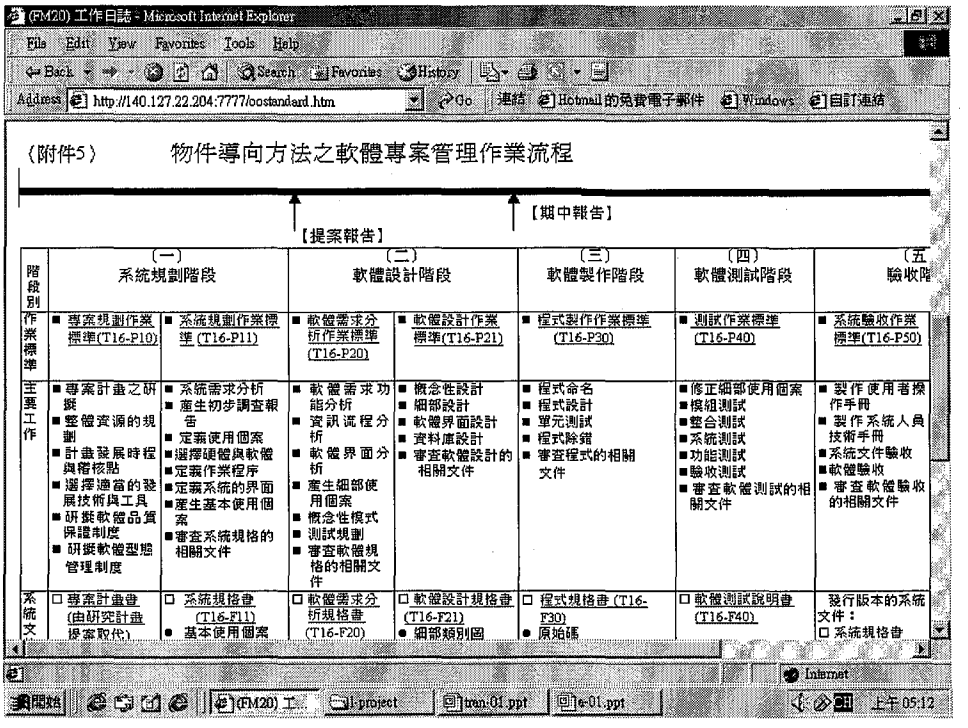


圖 13、物件導向方法之軟體專案制度之 Web 畫面

6. 發展雛形系統階段

此一階段主要工作是根據建立「物件導向方法之軟體開發作業流程與制度」發展一個「販賣機系統」的雛形系統。此一階段的工作重點是針對「物件導向方法之軟體開發作業流程與制度」，進行制度與流程的驗證工作。此一案例是以開發一個「販賣機系統」的專案計畫為對象，有關「販賣機系統」的所有系統文件與表格均一一予以詳列。軟體專案管理制度系統文件的範例：為驗證本研究計畫設計的制度與文件應用之可行性，以『販賣機系統』為例(附錄一)，將系統文件與表格的範例均建置在該系統中，此一雛形系統對於初次使用者而言是極為有用的參考資料。

伍、分析與討論

根據本研究的經驗，一個以使用者為需求導向設計之物件導向專案開發制度從整體觀點而言，學習時間較短而且制度趨於簡單化，導入的過程亦比較容易。本制度最大的特色是將整套制度建立在 Web 作業環境，使用者可以直接透過網路擷取相關作業標準，並予以應用在軟體專案的開發過程。對國內實務界而言，導入物件導向軟體專案開發制度的工作是一項新的嘗試，至目前為止，國內尚無文獻提出一套完整的作業程序，對於導入的單位而言，具有一定程度的風險。以本研究在A公司的經驗，從無任何標準作業制度到整個制度的建立完成為止，經歷規劃、分析、設計、建置與導入的過程，耗費的時間長達一年以上，而且投入的人力也高達二十個人以上。本研究亦發現使用者熟習物件設計的方法必須發費一段較長的學習時間，在導入初期並不易有立即成效。同時，物件導向的方法對於大型複雜軟體專案的開發工作助益較大，主要理由是大型複雜軟體專案具有一些共同元件(Component)，軟體再用性較高造成的效果。人力資源管理乃軟體專案管理一個相當重要的環節，為了增進軟體設計人員的生產力與品質，在軟體開發制度與流程方面，本研究將軟體開發階段的工作，制定一套作業標準。軟體設計人員可以透過 Web 作業系統直接取得相關的作業標準、標準文件格式。同時，在 Web 作業系統也提供一個販賣機系統的範例，對於使用者應有相當程度的助益。總之，本研究所發展的管理制度對於從事軟體開發工作者而言，提供了一個完整與標準化的作業環境，對於工作品質與效率的提升具有相當大的助益。同時，利用 Web 系統進行溝通活動也提升知識分享的作業品質，對於軟體專案管理的人力資源的運用頗有幫助。

導入新的制度以後，本研究彙總 A 個案公司的研發工程師的意見以後，歸納出下列幾點：1. 軟體設計品質方面—在系統規劃、分析與設計方面，均利用物件導向 CASE 工具，對於系統設計品質例如：完整性、正確性、一致性、系統架構等有相當程度的助益，2. 軟體開發時程方面—針對一項研究專題開發時程，平均縮短幅度約三分之一，3. 軟體設計生產力方面—使用新的 CASE 工具，軟體設計生產力提昇 2 倍左右，4. 軟體系統的維護工作方面—由於 Web 系統對於文件管理要求十分嚴格，負責系統維護工作者，及 5. 專案控管能力方面—由於所有的系統文件與專案進度列入管制，對於專案管理者進度

的掌控已經有顯著的改進，目前，每週的工作會報均利用本制度的資料進行跟催與討論，對個案公司的業務已經有顯著的效益。本研究係採個案研究法進行，在個案公司的寶貴的實證經驗與一些重要的發現，相信對於國內企業欲推行物件導向開發制度應有一些助益。

近年來，一些新的物件導向開發方法，例如：元件式開發方法(Component-Based Software Development, CBSD)、設計樣式(Design Pattern)與模式導向法(MDA)等被運用在一些複雜專案或大型軟體專案的開發工作。針對於上述的新開發方法，茲將本研究研發之物件導向開發制度擬採取的因應措施予以說明如下。在系統規劃階段，需求塑模過程，要求採用 Rational Rose 公司所發展的 CASE 工具，利用使用個案圖、活動圖等描述系統的功能。其次，在軟體設計階段，針對元件式系統塑模工作，本制度的規定的做法如下：1. 元件定義—對於一些具有獨特性與明確的功能之軟體元件，制定元件編號、名稱、規格、應用對象、基本功能、使用者等作明確的規範，2. 元件庫管理—元件儲存的基本資料、元件的專有名詞的資料(名詞、詞彙、規則、索引)、更新作業流程、元件的儲存方式、存放目錄區等，至於元件的組裝與再用性方面，本制度的規定的做法如下：1. 元件查詢—利用關鍵字或功能等，查詢元件庫的資料，若有適合的元件則列入再用的元件範疇，2. 元件的再用—對於現有元件且適合新的軟體開發需求的則予以修正或使用，3. 元件的組合與應用—針對於一些元件需要重新組合的作業流程的做法作完整與詳細的規範。

至於一個企業部門推行一個新軟體的開發制度，基本上需考慮導入公司的人力素質、公司的財務狀況、產品與服務的需求水準、工作品質的要求、技術能力、市場的競爭等多方面的因素。由於每一個企業本身的經營體質、技術能力、市場的環境與規模之差異性，在推動新軟體的開發制度過程應做一些調適的工作，茲將本研究的成果如何應用至其他的公司或產業方面，一些具體的建議如下：

1. 人員方面—軟體設計人員對物件導向的分析與設計能力，應是導入工作的的第一要務。建議導入的初期派遣一些學習能力較強的人員接受有關物件導向課程的訓練，培養技術移轉工作的種子人員，對未來技術的推廣的績效較佳。
2. 技術方面—選擇與評估一種適合的物件導向 CASE 工具是導入物件開發技術的重要工作項目，因此作業程序與工具的整合應列為考慮的重點，一方面提昇人員使用新技術的能力，另一方面，檢討與修正引進新工具的作業程序。
3. 作業流程方面—由於物件導向的開發方式與傳統的作業流程有相當大的差異，應重新檢討一些大型軟體專案或企業資訊系統的開發流程以符合物件導向開發方法的作業需求。
4. 建立制度方面—新制度的設計重點在於制度的彈性、標準化以及容易使用為原則。
5. 制度的推動與執行方面—高階主管的重視與支持與使用者的參與和支持是相當重要的，訂定獎勵制度以及定期的檢討與修正，對於推動的績效應有相當大的助益。
6. 財務支援方面—財務部門原則上應配合引進新制度的一些資金的需求提供一些必要的支援，但是為降低失敗的風險與達成經濟效益的目標，建議企業應配合引進的過程，進行階段性的投資，以減少初期投資需要龐大的資金的問題。

茲將本研究在個案公司導入專案管理制度的過程中，遭遇的問題與解決問題的經過情形整理於表 6。以下擬將本研究的成果與使用者對專案制度的需求加以說明如下。根據本研究的問卷資料，在使用者對專案制度需求方面，使用者重視的前三名分別為彈性、完整性與方便性，因此，制度設計方面應考慮的重點是制度內容的完整性、易於使用與制度修改的彈性。其次，導入新制度的過程方面，使用者重視的前三名分別為高階主管的重視與支持、導入軟體工具的配合與使用者的專業能力，因此，如何加強與高階主管的溝通，取得主管的信任與支持是推行工作的重點項目，同時，配合軟體工具的引進，進行人員的教育訓練工作，提昇軟體設計人員的專業能力對推動的績效應有相當大的助益。至於設計制度方面，使用者重視的前三名分別為標準化、人員的角色與工作以及軟體工具的配合度等，符合國際品質標準應被視為一項重要的考慮因素，如何與軟體工具配合以及詳細規定人員的角色與工作應是設計制度的重要工作內容。

表 6、導入物件導向專案開發制度的問題與解決的方法

問題與對策 / 主要構面	遭遇的問題	解決的方法
(一)人員方面	■ 對於物件導向設計的方法均十分陌生，經驗也相當欠缺，因此導入初期部份研究人員有恐懼心理 ■ 不願意改變目前的作法	■ 主管表現強烈企圖心與支持態度 ■ 不斷的進行溝通工作 ■ 對於表現優異的人員進行公開表揚與獎勵
(二)作業制度方面	■ 作業流程方面普遍是採用個人的經驗法則 ■ 缺乏標準化制度與流程	■ 制定一套符合ISO 9000-3之作業標準 ■ 配合物件導向的開發方法設計新的作業流程 ■ 將軟體工具與作業流程予以整合
(三)文件方面	■ 製作系統文件耗費許多的人力、時間與成本，製作系統文件意願很低 ■ 不了解系統文件包含的內容且製作系統文件的水準參差不齊	■ 利用CASE工具(Soda)自動產生系統文件 ■ 主管全力督導親自審核部屬呈閱之研發專題的系統文件 ■ 利用Web支援系統文件的標準格式
(四)技術方面	■ 研發工程師不熟習物件導向的設計方法 ■ 大部研發工程師僅利用一般性的語言工具(Visual Basic, C, C++, Java....)直接設計軟體 ■ 開發完成的應用系統之軟體進行移轉工作並不容易	■ 加強設計人員的分析塑模與設計的能力 ■ 加強UML的訓練 ■ 專案人員進行輔導工作 ■ 採小組共同學習方式 ■ 定期評估工程師的技術能力
(五)工具方面	■ 欠缺適當的軟體工具支援導致生產力低落	■ 進行CASE工具的評估與選擇 ■ 引進一些必要的軟體工具例如：進行分

	■ 軟體工具的種類相當多且各種功能並不相同 ■ CASE工具的價格相當昂貴	析與設計的CASE工具（Rational Rose 2001），專案規劃與控管的工具(Project 2000) ■ 安排許多主題進行實作練習以熟習軟體工具的操作

資料來源：本研究整理

綜合以上本研究的成果，企業在導入專案開發制度的過程中應注意下列幾項實施原則：

1. 爭取主管的信任與支持—首先，必須使主管充分了解推動專案開發制度的重要性，爭取主管的信任與支持是推動計畫成功與否的主要關鍵所在。
2. 鼓勵使用者的參與—如何有效的鼓勵使用者的參與，必須透過制度的運作與一些激勵措施，如果從使用者的觀點出發，設計的作業流程有助於提升工作的品質與生產力，使用者的參與意願就相當高。
3. 教育訓練工作是導入過程一項重要的關鍵性活動—使用者對於軟體工具的操作通常要經歷過一段長期的訓練過程，訓練並不是可以見到立竿見影的效果，仍必須透過實際問題的操作才能了解軟體工具的功能。
4. 作業流程之設計必須符合標準化與簡單化的要求—作業流程必須符合使用部門功能方面的需求，因此，必須簡化工作，力求作業標準化。
5. 配合作業制度的運作，導入一些適合的軟體工具—推動制度的運作時，必須有適合的軟體工具之支援才能發揮效果。
6. 減少使用者的負擔儘可能使作業朝向自動化—推動一項新的制度必須儘量減少使用者的作業負擔，減少使用者的抗拒心態，則制度的推展也較為順利。
7. 設計制度必須預留適當的彈性—制度設計並非完整，因此必須不斷修正，制度內容若能保留適當的彈性，可以減少修正過程的衝擊與影響。
8. 制度規劃工作必須以整體架構作基礎—設計制度若有一套完整的整體架構作基礎，對於制度的內容與品質將有十分大的助益。

陸、結論與建議

總而言之，本研究係整合物件導向設計方法、專案管理的觀念與 CASE 工具，發展一套以物件導向的軟體專案開發制度，提供業者自行發展軟體專案制度之一個基礎架構。基本上，本研究是從資訊整合應用的觀點，將物件導向的設計理念融入專案開發過程，建立一個完整的軟體開發模式，對於學術研究工作而言，揭開許多值得研究的重要課題，例如：如何將軟體工具與軟體開發制度整合、如何利用 CASE 工具進行軟體專案的開發、專案開發制度之建立與流程自動化等。其次，對實務界而言，本研究建立一套物件導向的軟體專案開發制度，可作為國內軟體部門自行發展與設計專案制度的參考依

據。對國內軟體業者而言，國外發展的專案開發制度均以大型軟體專案為主，可是台灣的軟體產業的規模以中小型居多，如果完全移植國外的制度並不一定適合國內業界的需求，因此必須考量國內產業的形態、市場規模、產品特性與軟體專案之需求等，重新設計一套真正符合國內軟體業者的專案管理制度，本研究即是朝向此一目標的具體成果之實現。

至於本研究的限制，茲說明如下：

1. 本研究的重點在於如何建立一個物件導向開發模式，為驗證本研究模式應用的可行性乃以國內一家大規模的研發部門作為實證研究的對象，至於在國內的企業進行大樣本的物件導向的制度的應用性調查研究，則不列入本研究的範疇。
2. 由於模式驅動架構方法論的應用目前仍處於快速發展階段，本研究僅針對未來發展之趨勢的因應做法作一些討論，並未做深入詳細的論述。
3. 由於個案公司的研究專題主要是以中型規模之軟體專案為主，大型軟體專案的案件則較少，因此本制度在大型軟體專案或相當複雜的大型科學研究性質的專案計劃之應用成效，仍有待後續研究予以澄清。
4. 由於個案公司的作業制度相當完整而且通過 ISO 9001 的驗證，因此在資訊系統的開發能力方面，擁有相當高的水準，因此本制度的推動以符合資訊能力成熟度模式(CMM)水準 Level 2 以上的企業為主，預期推動的成效方面應有較佳的表現。
5. 本研究的重點在於物件導向開發制度的建立與如何將發展的制度導入至企業內部的工作上面，主要研究工作重點是根據物件導向的開發作業需求建立一套完整的規範、作業流程與制度。至於在個案公司發行問卷的目的在於了解使用部門人員對於設計的需求，因此本研究發行問卷的時間點乃選在需求分析階段，至於實際導入一段期間以後產生具體效益，建議採用質性研究方法深入觀察實際推動以後的成效，本研究並未將此列入研究的範疇。
6. 研發部門的人力素質相當高，碩士學歷的人員佔大部分，因此對於新技術的接受意願與學習能力相當高，對推動的成效在有相當程度的助益，至於在一些制度不完整或人力素質較低的中小企業推行本制度，尚有待作進一步的實證研究。

瞻望未來，一個企業要擁有較佳的競爭優勢，必須不斷進行各項創新活動，例如改進製程、降低生產成本、研發新技術與新產品。因此企業研發部門係居於一個領導創新活動的重要角色，不論是新產品或製程的新技術研發，必須有效的累積研發的技術與經驗，其中軟體開發工作乃扮演一個關鍵性角色。因此，對A公司而言，如何將研發工作的軟體開發過程進行系統化的管理與全面提升軟體設計人員生產力，乃是當前的工作重點，因此軟體專案管理制度與流程的建立被列為A公司一項優先執行的重點工作。茲將本研究的主要貢獻敘述如下：

1. 建立一套完整的軟體專案管理制度與作業流程。
2. 建立一些重要的軟體開發過程的標準文件與管理制度，並且利用檔案伺服器將系統文件集中管理，如此一來可以累積專家的經驗與知識，而達到知識再用的效果。
3. 建立一個 Web 的作業環境，軟體開發人員可以隨時透過網路的傳輸與資訊交流，建立一個完整的技術資訊系統。

4. 配合制度的建立過程，規劃最新的軟體開發作業環境與軟體工具，作為建立專案開發工作平台之參考依據。.

綜合以上所述，茲將本研究的經驗與心得彙整如下：1. 建立一套完整的專案開發作業流程是研發部門累積知識的重要基礎，唯有透過標準作業程序，才能建立實用與完整的技術資料庫，2. 導入新資訊科技，必須作好溝通與教育訓練的工作，才能事半功倍，3. 制度設計應力求簡單化與實用性為第一考量，以及4. 配合管理制度，導入CASE工具。諺云：「工欲善其事，必先利其器」，唯有工作簡化才能減少使用者的抗拒心態，導入新的資訊科技才能使專案制度的推行發揮預期的效果。由於本研究的實證對象是國內一家大規模的鋼鐵公司研發部門，雖然本研究已經對與個案公司進行了深度的分析與探討，至於本研究的實際成果是否能適用到其他公司或軟體業者，仍有待後續研究之進行，此乃本研究的限制。根據本研究的經驗，建議後續研究可以朝下列方向進行：1. 軟體部門導入大型物件導向軟體專案制度之實證研究，2. 軟體部門導入物件導向軟體專案制度之成功關鍵性要素之探討，3. 整合性軟體開發環境之建立，包括軟體開發制度、流程與軟體工具之整合，與 4. 模式驅動法之物件導向軟體專案制度之建立。

致謝

作者感謝審稿委員提供許多寶貴的意見，並感謝國科會專題研究計劃的補助，編號：NSC90-2416-H-020-006

參考文獻

1. 謝遠敦、林信惠、黃明祥，民 86，「動態軟體專案管理之研究」，第八屆國際資訊管理學術研討會論文集，台北：政治大學， pp. 586-593。
2. 林信惠、陳俊賢、黃明祥、李坤清、吳裕光，民 84，「我國資訊軟體業推動 ISO 9000 之研究」，第六屆國際資訊管理學術研討會論文集，台北， pp. 470-477。
3. 李坤清，民 83，軟體專案管理模式研究，中山大學企業管理研究所博士論文。
4. 王慶富，民 85，專案管理，台北：聯經出版社。
5. Abdel-Hamid, T. k., "The Slippery Path to Productivity to Improvement," IEEE Software, 13(4), 1996, pp.43-52.
6. Ambler, S.W., "Agile model Driven Development is Good Enough," IEEE Software, Vol. 20, Issue: 5, Sept.-Oct. 2003, pp.71-73.
7. Baker, E. R. and Fisher, M. J., "Software Quality Program Organization," Edited by Schulmeyer, G. G. and McManus J. I., Handbook of Software Quality Assurance, Second Edition, Van Nostrand Reinhold, 1992, pp. 49-74.
8. Bennett, Simon, McRobb, Steve, and Farmer, Ray, Object-Oriented Systems Analysis and

Design Using UML, McGraw-Hill International Editions, 1999.
9. Berkeley, D. Hoog, R. De, and Humphreys, p., Software Development Project Management-Process and Support, Ellis Horwood, New York, 1990.
10. Boehm B. W., Software Engineering Economics, Englewood Cliff, NJ: Prentice-Hall, 1981.
11. Boggs, M. and Boggs, W., Mastering UML with Rational Rose, SYBEX Inc., 1999.
12. Bruce, P. and Pederson S. M., The Software Development Project — Planning & Management, Song Kung, 1982, p. 50.
13. Cooper, J. D. and Fisher, M. J., Software Quality Management, Petrocelli Books, Inc. 1979.
14. Cleland, D. I., William R. King, Project Management Handbook, Van Nostrand Reinhold, New York, 1988.
15. France, R., Chosh, S.; Song, E., and Kim, D.K., "A Metamodeling Approach to Pattern-based Model Refactoring," IEEE Software, Vol. 20, Issue: 5, Sept.-Oct. 2003, pp.52-58.
16. Huang, M.S.，Lin, H.H., "Integrating ISO 9001 and the CMM in Software Development Process," Proceedings of The Seventh International Conference on Comparative Management, National Sun Yat-sen University, 1996, May, pp. 122-131.
17. Huang, M.S.，Lin, H.H., "An Empirical Study of Software Quality Assurance with Case of On-line Responses Time Improvement," 第七屆國際資訊管理學術研討會論文集，中原大學，民國八十五年五月，pp. 447-457。
18. Humphrey, W. S., Managing the Software Process, Addison-Wesley Pub. Co., New York, 1989.
19. Jacobson, I., Booch, G., and Rumbaugh, J., The Unified Software Development, Addison-Wesley Pub. Co., 1999.
20. Kan, S. H., Dull, S. D., Amundson, D. D., Lindner, R. J., and Hedger, R. J., "AS/400 Software Quality Management," IBM Systems Journal, Vol. 33, No. 1, 1994, pp. 62-87.
21. Kan, S. H., Basili, V. R., Shapiro, L. N., "Software Quality: An Overview from the Perspective of Total Quality Management," IBM Systems Journal, Vol. 33, No. 1, 1994, pp. 4-18.
22. Liu, L.C., and Horwitz, E., "A Formal Model for Software Project Management," IEEE Transactions on Software Engineering, Vol. 15, No., 10, Oct. 1989, pp. 1280-1293.
23. Phillip, B. and Pederson, S. M., The Software Development Project, Planning & Management, 台北： 松崗，1982。
24. Pressman, R. S., Software Engineering — A Practitioner's Approach, Fifth Edition, McGraw-Hill, Inc., 2001.
25. Rakos J. J., Software Project Management for Small to Medium Sized Projects, 1990, pp.145-160.
26. Rumbaugh, J., Jacobson, I. and Booch, G., The Unified Modeling Language Reference

Manual, Addison-Wesley, Inc., 1999.
27. Schmauch, C. H., ISO 9000 for Software Developers, ASQC Press, 1995.
28. Uhl, A., "Model driven Architecture is Ready for Prime Time," IEEE Software, Vol. 20, Issue: 5, Sept.-Oct. 2003, pp.70, 72.
29. Wideman, R. Max, A Framework for Project and Program Management Integration, PMBOK Handbook Series Vol. No. 1,West Cost British Columbia PMI Chapter, 1991, p. VII-5.
30. Yourdon, E., Decline & Fall of the American Programmer, Prentice Hall Building, 1992.
31. http://www.omg.org/mda/mda_files/Model-Driven_Architecture.pdf.

附錄一

【販賣機系統之需求分析規格書的範例】

1. 導論

1.1 目標

自動販賣機系統主要是一項提供顧客購買飲料之用途自動化設備，為了達到全自動販賣的銷售功能，本系統必須具備一些與銷售人員的販賣能力相近之功能，例如提供顧客投幣、選擇飲料以及找零錢等功能。茲將本系統的目標說明如下：(1)具備一般產品的銷售能力，可以完全取代銷售人員而達到無人化銷售的目標，(2)記錄自動販賣飲料之相關資訊，作為補充存貨的基本資料來源，(3)簡單的系統操作能力，任何人均能輕易的操作，不必再學習，與(4)防止不良份子破壞販賣機的設備，同時具有通報警方的能力。

1.2 範圍

自動販賣機系統的作業範圍定義如下：(1)溫度控制，(2)硬幣的檢測與計算，(3)存貨狀況的顯示，(4)使用者操作方式的處理，(5)自動顯示使用者可以選擇的飲料種類，與(6)系統異常狀況的處理。

1.3 功能摘要

自動販賣機系統主要是提供顧客購買飲料之用途，為了達到全自動販賣的銷售功能，系統必須具備一些與銷售人員的販賣能力相近之功能，包括： (1)溫度控制－設定飲料冷凍的溫度，(2)操作界面控制－投幣口、退幣桿、飲料選擇按鈕、金額顯示面板、飲料顯示燈，(3)金額計算－鑑別硬幣真偽、硬幣面值的大小、累計投入硬幣的金額、顯示累計金額、找零錢，(4)存量計算－飲料存量之扣減、存貨有或無之顯示，(5)安全系統的控制－防止偽鈔、防止販賣機被人毀損或物品被竊盜的功能。

1.4 假設與限制

一般而言，自動販賣機系統是在一個開放性的公共場所中使用，最大電力負荷是 150 伏特。

2.軟體需求的描述

2.1 功能的需求（細部使用個案）

*選擇鈕

(1) light_on(燈亮)

由物件 MACHINE 之 display_amout(金額顯示)傳來顯示之金額訊息

◎ If 顯示之金額扛 飲料選擇鈕之金額 then 飲料選擇鈕燈亮

◎ 將"飲料選擇鈕燈亮"訊息傳給物件 MACHINE(販賣機)

(2) light_off(燈熄)

◎ If 啟動本程式 then 將所有飲料選擇鈕均歸零至燈熄

◎ If 由物件 MACHINE'之 push_button(按鈕)傳來按鈕訊息 then 將所有飲料選擇鈕均歸零至燈熄

(3) display_empty(售完顯示)訊息

◎ 由物件 CACULATOR(存量計算器)之 decrease(遞減)傳來該飲料存量 =0 之訊息

◎ 該飲料之售完燈亮

*退幣桿

(1) If 顧客拉動退幣桿 then 將物件 MACHINE 之 display_amout(金額顯示)所顯示之金額全數退還

*顧客

(1) 將退幣訊息傳給物件 MACHINE 之 display_amout(金額顯示) Use Case Diagram

*金額計算器

(1) increase(累加)

◎ 由 MACHINE 物件之 insertcoins(投幣-接受)傳來訊息

◎ 判斷錢幣金額

◎ amount 累加

◎ 將 amount 之結果傳給物件 MACHINE 之 display_amout(金額顯示)

(2) change(找零)

◎ 由物件 MACHINE 之 push_button(按鈕)傳來顧客按鈕訊息

◎ 結算找零金額

◎ 輸出找零錢幣

◎ 將找零完畢訊息傳給 Reset(重置)

(3) reset(重置)

◎ If 由 change(找零)傳來找零完畢訊息 OR 由物件 BAR(退幣桿)傳來退幣訊息 then 將顯示金額歸零並傳給物件 MACHINE 之 display_amout(金額顯示)

*存量計算器

(1) decrease(遞減)

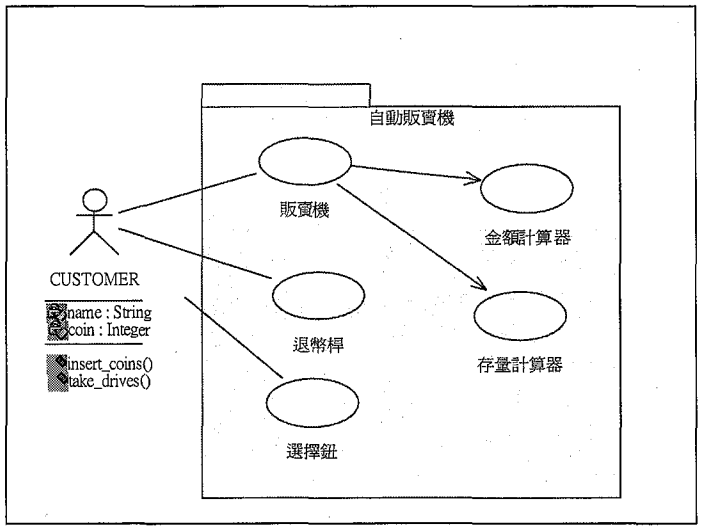
◎ 由物件 MACHINE(販賣機)之 push_button(按鈕)傳來顧客按鈕訊息

◎ 該飲料存量減一

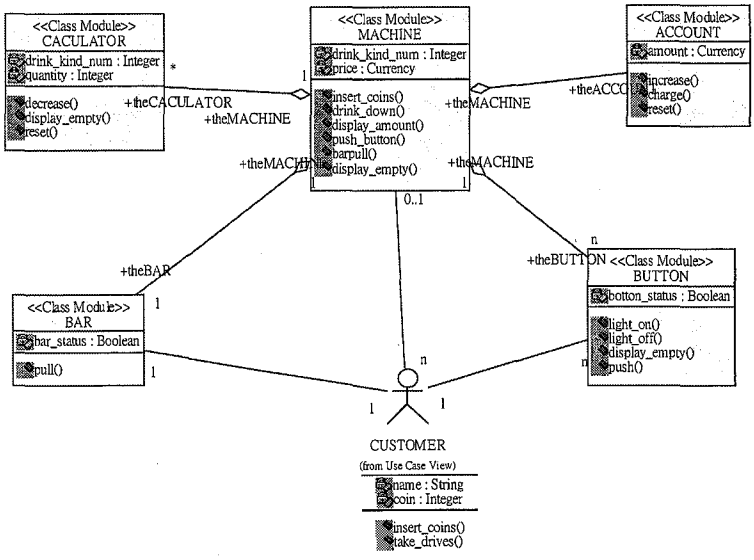
◎ If 該飲料存量 = 0 then 將訊息傳給物件 BUTTON(選擇鈕)之 display_empty(售完顯示)

(2) reset(重置)

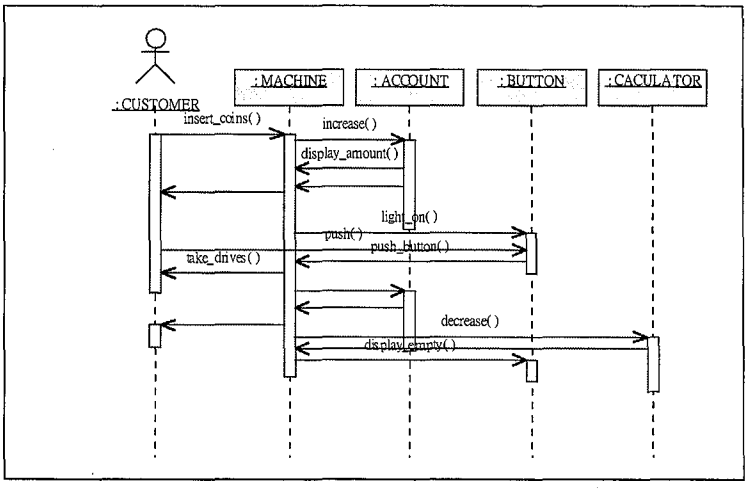
◎ If 啟動本程式 then 將所有飲料存量均歸零至起始值



2.2 類別圖



2.3 類別之間訊息傳遞的需求（互動圖、序列圖或合作圖）



2.4 人機界面的需求

設計一人機介面,期與真實販賣機之介面相似,主要模擬真實的販賣機來實際實作.

2.5 外部通訊的需求

販賣機需與伺服器(server)連線,經由伺服器可得知販賣機的販賣飲料數量,庫存數量

2.6 其他特殊的需求

(無)

3.驗證準則

3.1 軟體功能的績效需求

主要是包括系統作業的效率,主要不能反應太慢,如按完按鈕時10秒,飲料就要掉出,不能讓顧客等太久,否則其效能差,則將降低顧客的購買慾

3.2 軟體功能的驗證程序

其主要的測試程序:

1. 投錢時看系統是否能辨別出,$10 $5 $1 等,同時檢驗其金額計算器是否有增加或減少及是否有溢位.
2. 在一定金額下,檢驗是否可按的按鈕有亮,或是不該亮的也亮了.
3. 再選擇飲料時,檢驗其存量計算器是否有減少或增加,飲料是否有退出,飲料是否一樣.
4. 最後,檢驗此系統是否有回覆到初始狀態

3.3 其他特殊的驗證需求

(無)

4.附錄

(1)附錄 A：相關系統文件

□ 專案計畫書

□ 系統規格書

(2)附錄 B：同義語與縮寫字

□ 專案管理(Project Management, PM)

□ 人機界面(Human Computer Interface, HCI)

(3)附錄 C：定義與命名

□ 顧客---是個別的消費者,持有硬幣,擁有飲料及交易成交與否之選擇權

□ 金額計算器---將投入硬幣依序完成累加,並不斷將結果傳回販賣機顯示

□ 存量計算器---販賣機掉出飲料後,存量計算器則將該種飲料的存量遞減,同時並判斷是否為 0,若售完燈亮時,顯示無存量,否則繼續備用狀態

□ 退幣桿---在顧客按下選擇鈕完成交易前,若拉動退幣桿則將所有金額退還顧客

□ 販賣機---於顧客投入硬幣時,判斷硬幣是否有效及金額的值,加總後顯示,待顧客按下選擇鍵後掉出飲料

□ 選擇鈕---不同飲料有不同價格,待金額計算器累加到該飲料售價,則該選擇鈕燈亮起,顧客按下後將選擇傳回販賣機