CHAPTER

1

系統發展環境

Part I

本章目標

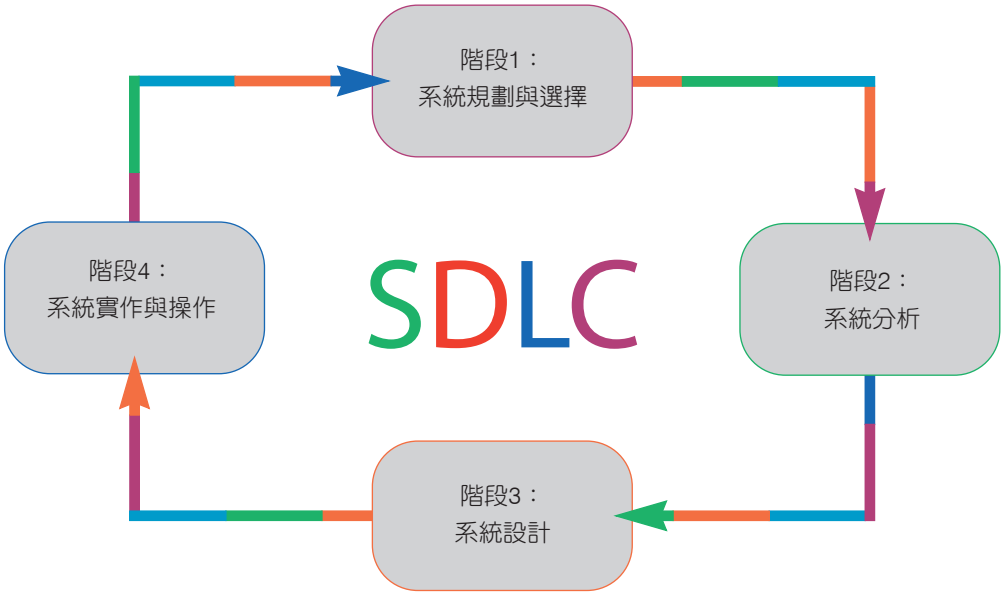
學完本章後，你應該能：

- 定義資訊系統分析與設計
- 探討結合系統程序與資料觀點的現代系統分析與設計方法。
- 描述資訊系統發展過程中，系統分析者在組織中所扮演的角色。
- 描述不同的軟體來源。
- 描述資訊系統的三種類型：交易處理系統、管理資訊系統和決策支援系統。
- 描述資訊系統發展生命週期（SDLC）。
- 列出系統發展生命週期的選擇方案，包括描述電腦輔助軟體工程（computer-aided software engineering, CASE）工具在系統發展中所扮演的角色。

本章簡介

資訊的蒐集、組織和詮釋往往是商業決勝的關鍵。系統分析與設計是一套經過驗證的方法，透過這套方法能發揮資訊的最大效能，以協助大型或小型企業的經營。在組織中，跟系統分析與設計關係最深的就是系統分析者，這份職業往往能讓人從提升電腦和人際關係技巧的豐富歷程中，獲得極大的樂趣。 系統發展生命週期（system development life cycle, SDLC）是一個有效資訊系統發展的中心。我們將著重於 SDLC 的四個重要階段；規劃與選擇、分析、設計、實作與操作。值得注意的是這些階段是會隨著組織的目標而變。圖 1-1 是 SDLC 的圖示。書中的每一章都會包含一個新的 SDLC ，並說明它涵蓋了哪些階段。 這部分的內容提供資訊系統課程的入門簡介，讀者最好對電腦資訊系統已經有些基本的認識。第一章先介紹一下系統分析，並為本書後續的內容建立基礎。

圖1-1
系統發展生命週期（SDLC）的四個階段：（1）規劃與選擇（2）分析（3）設計（4）實作與操作



1.1 什麼是資訊系統分析與設計？

資訊系統分析與設計（information systems analysis and design）資訊系統發展與維護的過程

資訊系統分析與設計這套方法最早是由 IBM、Pepsi 和 Hasbro 等公司發展出來的，用來建構一套具有基本商業功能且方便維護的資訊系統，希望能夠執行如追蹤顧客的姓名與地址、下訂單、和支付僱員薪資等功能。系統分析與設計的主要目的在於提升組織系統效能，為了達到這個目標，通常會使用軟體的來幫助人員更輕鬆而有效的達成工作上的要求。系統分析者是軟體發展工作的靈魂人物。資訊系統的分析與設計工作建立在：

網路搜尋
關於資訊系統與新技術每年新增的名詞，請造訪http://www.prenhall.com/valacich取得所有相關主題。

- 你對組織的目標、結構與運作的了解程度。
- 你具備如何利用資訊技術以提高工作效益的專業知識。

為了儘可能提高工作的成效，你必須依循一套有架構的運作方式。圖 1-1 所示的 SDLC 是一套分成「確認（identifying）」、「分析（analyzing）」、「設計（designing）」、和「實作（implementing）」四階段來處理資訊系統的方法。全書會以 SDLC 來統整對系統發展流程的探討。在談到 SDLC 之前，我們先說明一下何謂系統分析與設計。

1.2 系統分析與設計：核心觀念

應用軟體（application software）
用來支援組織中處理資料和支援使用者所設計的軟體，例如試算工作表、文件處理軟體與資料庫管理系統。

系統分析與設計的主要目標在於提昇組織的系統，而這項工作通常涉及**應用軟體**的發展或需求，以及對人員使用這些軟體的訓練。應用軟體，也稱為「系統」，是被設計用來支援特定的組織功能或工作流程，例如庫存管理、人事薪資、或是市場分析。應用軟體的目的是將資料轉換成資訊：比方說為書店的庫存部門所發展的軟體，就得能夠持續追蹤最新暢銷書的庫存量；為人事部門所發展的軟體，就得能夠持續追蹤員工薪資的變化。在坊間可以買到各式各樣的商用軟體，包括 WordPerfect、Lotus 和 PowerPoint 等，但是這些商用軟體未必適用於特殊的組織，因此組織有必要發展屬於自己專用的產品。除了應用軟體外，資訊系統還包括：

- 能執行應用軟體的硬體設備與系統軟體：系統軟體能幫助電腦運作，而應用軟體則是幫助使用者執行工作，譬如撰寫報告、準備傳單海報、和連結網際網路等。

- 文件與訓練教材：這些由系統分析者製作的教材能幫助員工使用他們協助開發的軟體。
- 掌控系統整體運作的專職角色：如啟動電腦並且讓軟體運作的人。
- 控制：軟體中用來防止舞弊和竊取的部份。
- 那些為了做好工作而使用軟體的人們。

以電腦為基礎的資訊系統應用，其組成元件整理成圖 1-2。系統整體所涵蓋的所有範疇中，我們特別強調應用軟體的發展－也是一個系統分析者的首要責任。

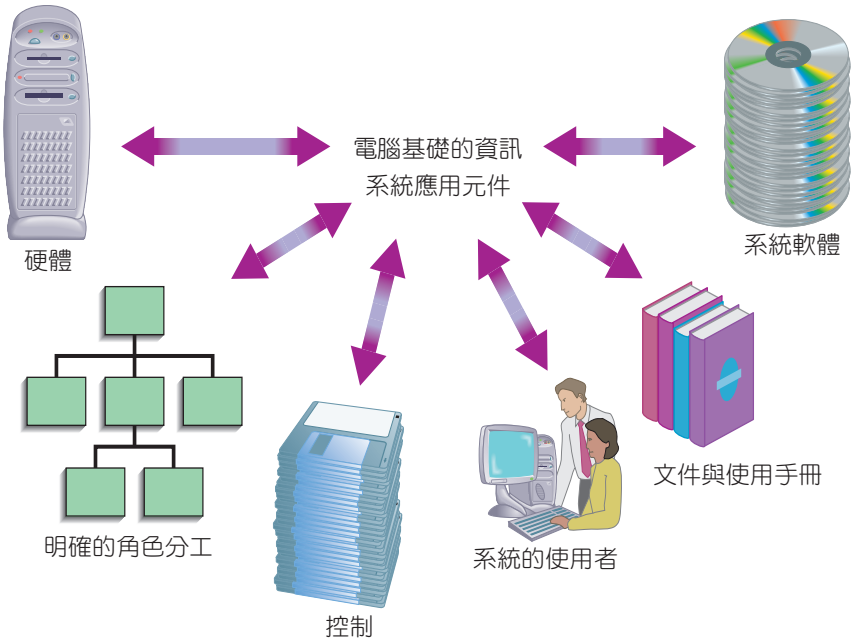
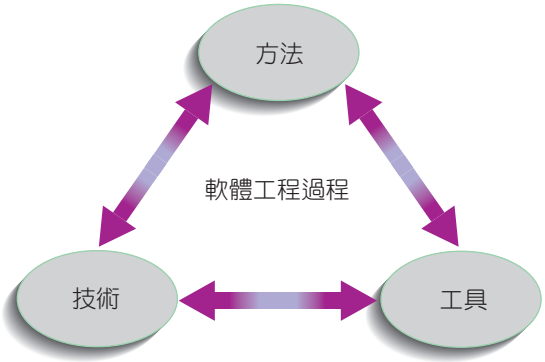


圖1-2
以電腦為基礎的資訊系統應用元件

我們的目的是幫助你了解資訊系統，並引導你依循軟體工程的流程，建立出理想的系統。軟體工程流程的核心（也是本書的核心）如圖 1-3 所示，包括驗證過的方法（methodologies）、技術（techniques）和工具（tools）。方法（methodologies）是一系列的處理步驟，用來協助你發展你的最終產物：資訊系統。大部分的方法都整合了多種的發展技術，如直接觀察和訪談目前系統的使用者。 技術（techniques）是指操作流程，一個分析師為了確保工作能順利進行、完成並能讓其他在專案團隊的人理解，因此必須制定

出標準的操作流程。技術能對廣泛的工作提供支援，包括對目前和未來的資訊系統使用者進行徹底的溝通，來決定你的系統該做些什麼，設計並管理系統發展計畫中的活動，規劃系統如何運作，以及設計你的系統所產生報表讓使用者查核他們的工作，如發票。工具（tools）就是電腦程式，如電腦輔助軟體工程（computer-aided software engineering, CASE）工具，它可以讓專業技術的使用更加容易。方法、技術和工具這三個元素共同運作，形成一個組織著手系統分析與設計的門徑。

圖1-3
利用方法、技術和工具的軟體工程過程



在本章後續的內容中，你會學到系統發展的方法－資料導向法與程序導向法。你也能分辨出各種發展系統的人，以及他們所發展出的各類型系統。在本章的最後，我們會討論支援系統發展過程的一些方法、技術與工具。在我們要進一步談論電腦資訊系統之前，先簡要討論一下所謂的「系統（system）」是什麼。

1.3 系統

這本書裡最常出現的名稱就是「系統」，瞭解系統與它們如何運作是瞭解系統分析與設計的重要關鍵。

1.3.1 系統與其組件的定義

系統（System）
針對商業功能，具有可識別的界限，為了某一目的而結合的一群相關程序。

系統是相互關連的商業程序、步驟（或元件）的集合，被使用於商業個體中，並為了一些目的而整合。例如，在薪資部門的系統主要是記錄支票的流通，而庫存管理系統則是管理庫存量，兩個系統是完全獨立的。一個系統應具備九種特性，其中的七個顯示於圖 1-4 中，每個特性的詳細說明如下所示。但從圖中可以看到系統存在於

一個較大的領域，那就是環境。系統與環境兩者之間的區隔在於，系統是由外部取得輸入的資料並處理之，然後再把結果輸出至環境中。圖中的箭頭顯示系統與系統外在環境的互動。

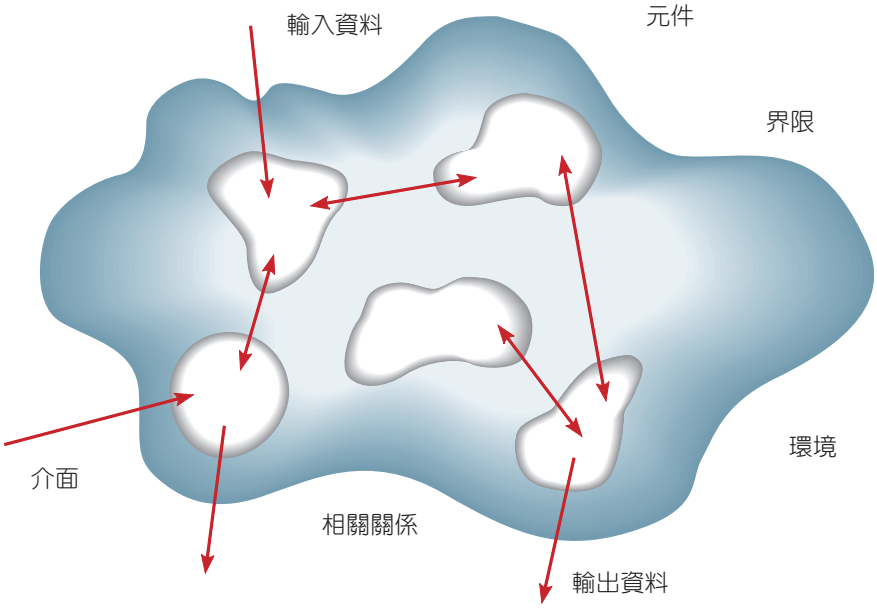


圖1-4
系統的七個特性

1. 元件（Components）
2. 相關元件（Interrelated components）
3. 界限（Boundary）
4. 目的（Purpose）
5. 環境（Environment）
6. 介面（Interfaces）
7. 輸入資料（Input）
8. 輸出資料（Output）
9. 限制（Constrain）

系統是由許多**元件**所組成。元件是不能再減縮（irreducible）的部份或組合，所以又稱為子系統，這個簡單的觀念對元件而言是相當有用的。舉例來說，就像汽車或立體音響的系統經過了適當的設計後，我們能藉由更換單獨的元件來修理或改善該系統，而不用修改整個系統。元件間是具有**相互關連性**的，也就是說，元件中的一個功能會以某種方式維繫著其它元件的功能。例如，某個元件的工作可能是產生消費者訂單的日報表，在諸如排序消費者訂單的日期等

元件（components）
元件是建構系統不能再減縮的部份或組合，所以又稱為子系統

相互關連性（interrelated）
系統的一部分與其它一個或多個系統部分的關連性

界限（boundary）
系統內部與外部的界線，用以區分一個系統與組織中的其它系統

目的（purpose）
系統的整體目標或功能

環境（environment）
系統外部與系統互動的任何事物

介面（interfaces）
系統與環境或與子系統的交接處

這類其它元件的工作尚未完成之前，該元件的工作就可能無法順利的進行。系統都會有一個**界限**，這個界限的內部包含所有的元件，並建立系統的界線，把它與其它系統分隔開來。界限裡的元件能被改變，反之，在界限以外的系統部份卻是不能被改變的。全部的元件是為了成就較大系統的整體**目的**而一起運作，這是系統存在的原因。 一個系統存在於**環境**之中，環境是指系統界限外面會影響系統的所有事物。例如，州立大學的環境包含未來的學生、基金、基金機關，以及新聞媒體。通常系統會與他所在的環境互動。大學會藉由開放參觀日和當地高中學校的招生活動與未來的學生互動。資訊系統會藉由取得資料（原始的事實）和資訊（有用的資料處理格式）來與他所在的環境發生互動，圖 1-5 說明如何把一所大學看成一個系統。系統與環境的交接處被稱為**介面**，而子系統和子系統之間也有介面。

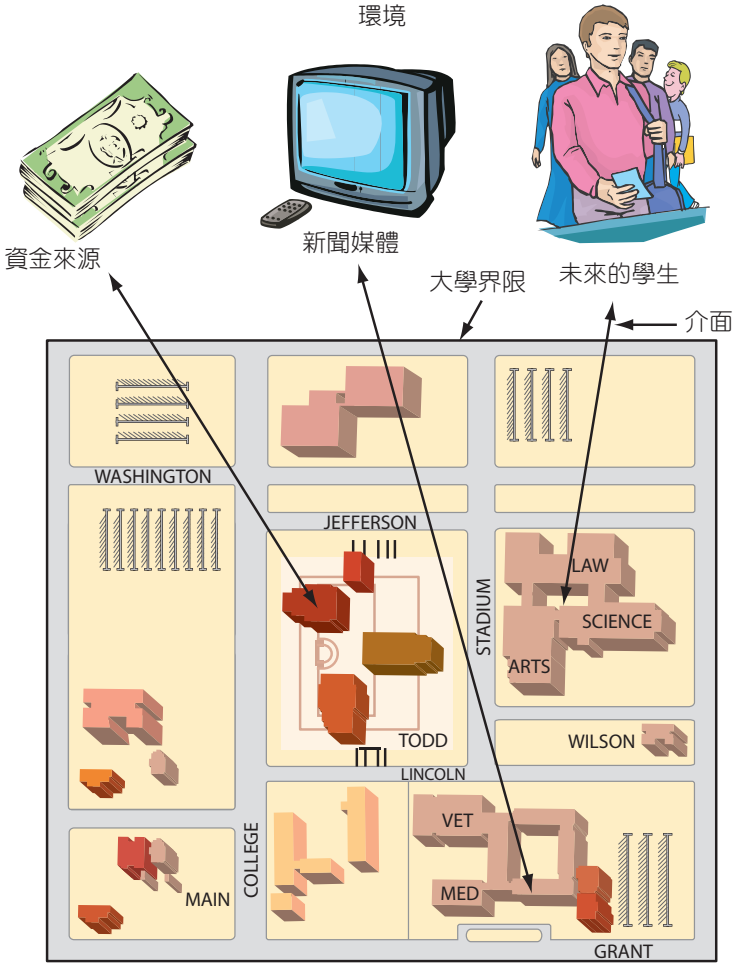


圖1-5
一所大學就是一個系統

系統必須面對它功能上的**限制**，因為系統所處的環境會對系統的能耐與達到目標的方式產生一些限制（從容量，速率，或功能的角度考慮）。有一些限制在系統裡面被加強（例如，可利用員工數目是有限的），而其它的限制在環境中被加強（例如，到期時間或規則）。系統為了它的功能而從所在的環境中取得輸入，就像人們從環境中吸收食物、氧氣以及水做為輸入一樣。如果有人在電梯內抽煙，吸取新鮮空氣的條件就被限制。最後，系統將功能執行後的結果輸出給它所在的環境，以達成系統的工作任務。當電力被切斷時，系統也勢必受到限制。

限制（constraint）
系統可達成的範圍

網路搜尋
了解系統的意義是成為系統工程師的基礎，請造訪http://www.prenhall.com/valacich取得所有相關主題。

1.3.2　重要的系統概念

有數個不同的重要系統概念是系統分析者需要知道的：

- 分解（Decomposition）
- 模組性（Modularity）
- 耦合（Coupling）
- 凝聚（Cohesion）

分解是把系統分解成較小元件的過程，這些元件本身也可能是系統（子系統），也能夠同樣的分解成更小的元件。分解為何有助於瞭解整個系統呢？比起較大且錯綜複雜的部分，較小與較簡單的部分是容易瞭解的多。而且分解一個系統將允許我們把注意力集中在系統的某個特定部分，方便我們改變整個系統中某個獨立的部分。分解是一種技巧，讓系統分析者：

分解（Decomposition）
將系統分解成較小元件的過程，也就是功能性分解

- 把系統變得較小、容易操作與易理解的子系統。
- 每次只集中注意力在一個範圍（子系統），沒有其它部份的干擾。
- 專注於系統的特定使用群組的部份，免除不必要的細節混淆使用者。
- 在不同的時段建構不同的系統部分，並有不同分析師的幫助。

圖 1-6 顯示一個可攜式 MP3 隨身聽的分解，系統分解為子系統來展示系統內部的運作。你可以將 MP3 隨身聽至少分解為三個實體子系統（註：同樣的 MP3 隨身聽分解邏輯子系統會產生不同的子系

統）。一個子系統的電池供應所有系統運作上所需電力。第二個子系統是一個儲存系統，它將許多 MP3 記錄儲存在硬碟。第三個子系統是子系統控制器，由一個印刷電路板所和不同的晶片所組成，它控制所有的紀錄、錄放和存取功能。將系統分解成子系統成為它的元件，將更能視察系統內部的運作，並且大大地提高我們對系統整體運作的認知。**模組**是分解後的直接結果，將系統分割成相當數量或比例均勻的模組。模組能簡單地表示系統，使系統更容易了解、重新設計與重建。例如，在圖 1-6 是 MP3 隨身聽的子系統模組，它顯示了分解可以讓全部系統更容易被瞭解。

模組（Modularity）
分割系統成相當數量或比例均勻的模組

圖 1-6
MP3 隨身聽的系統，包含電力供給、儲存和一個子系統控制器。

耦合（Coupling）
表示子系統間彼此相依的關係

耦合表示子系統間彼此相依的關係。子系統應是儘可能的獨立，因為如果有一個子系統失效，而其它子系統又非常依賴它時，其它的子系統也會失效或是在運作上出問題。從圖 1-6 看來，我們能說 MP3 隨身聽的元件具有緊密的耦合，由晶片與印刷電路板組成的控制系統是個最好的例子。MP3 隨身聽的每個功能是由晶片與電路板驅動，當有一部份的電路板發生故障，這表示要替換整個板子而不是試圖去釐清板子哪裡出問題並且修理它。雖然修理 MP3 隨身聽電路板是可行的，但是很沒有效益，找出問題修理它也許是值得的，可是這個工作所花費的價值遠低於電路板本身的價值。在家庭立體音響系統裡，元件間的耦合是鬆散地，因為子系統（例如，擴音

機，揚聲器與 CD 播放機）實際上是有完全分開獨立的功能。如果家庭音響系統裡的揚聲器壞掉的時候，只需要修揚聲器即可。

凝聚是指子系統只執行單一功能。在 MP3 隨身聽的例子裡，電力供給就是一個單一功能。

凝聚（Cohesion）
是系統或子系統執行單一功能

對系統作這番簡明扼要的討論，有助於你去思索電腦資訊系統，以及了解這些系統是如何被建構的。許多應用於系統的共通原則，一般也適用於資訊系統上。在下一節中，我們將回顧資訊系統的發展過程，以及支援資訊系統變革超過十餘年的工具。

1.4　現代的系統分析與設計方式

今日，系統的發展著重於系統整合。系統整合是讓來自於不同的賣方的硬體和軟體可以為了一項應用工作而共同運作。它也可以讓一個由程序性語言發展出來的系統，在一個由視覺化程式設計環境建構的新系統中運作。開發者利用視覺化程式設計環境，如 Visual Basic，設計出運作於主從架構平台（client/server platform）的使用者介面系統。在主從架構環境中，一些軟體執行於伺服器端，而較有效率的電腦通常被設計來讓許多人在上面取得軟體和儲存資料。另一些軟體則執行於客戶端的機器上，客戶端的機器是用於工作的桌上型個人電腦。資料庫通常存在於伺服器端上，其關係顯示於圖 1-7。網際網路也被組織成主從架構的格式。透過家用 PC 的瀏覽器軟體，你可以從世界不同的電腦中取得檔案與應用程式。因此，你的家用 PC 是客戶端，而網際網路上的所有電腦都是伺服器端。

組織也可能購買像 SAP（Systems, Application, and Products in Data Processing，系統、應用、與資料處理產品）或 Oracle 公司的廣泛型企業系統（enterprise-wide system）。廣泛型的企業系統是巨大而複雜的系統，是由一連串獨立的系統模組所組成。開發人員藉由選擇和執行特殊的模組來組合系統。廣泛型的企業系統通常包含了支援多種組織內部不同工作的軟體，而非只有一個或兩個功能而已。例如，一個廣泛型的企業系統可以將全部人力資源的管理、薪資、津貼與退休的功能，放在單一而整合的系統中處理。 現在有越來越少的企業願意去開發自家的系統。以下章節會介紹各種不同的

資訊系統技術來源。當我們想外購系統時，會考慮到有許多的軟體製造商。首先，無論如何你必須瞭解到在這個系統發展中，你將扮演的角色。

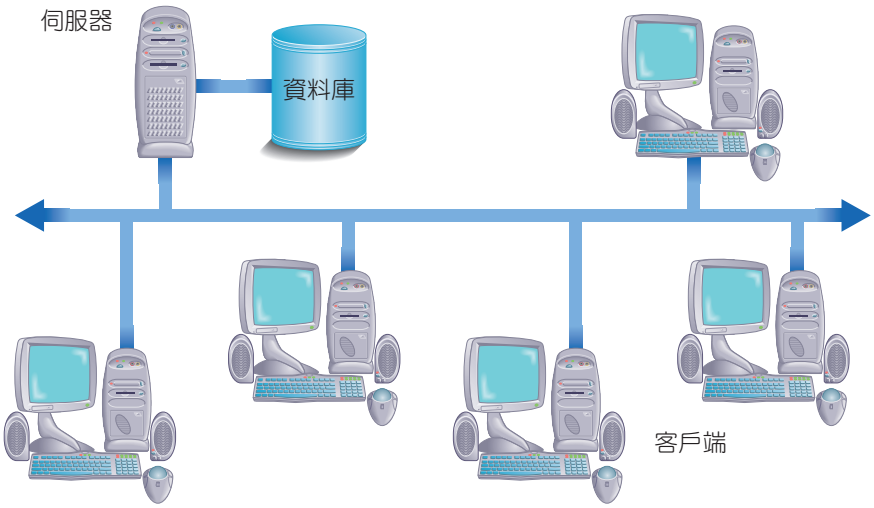


圖 1-7
主從架構模式

1.4.1 你在系統發展中的角色

雖然組織裡有許多人參與系統分析和設計的工作，但是系統分析者卻是要負最大責任的人。系統分析的工作會明顯而直接的影響組織的運作。無論是大公司還是小公司，這都是快速成長且有價值的職位。美國資訊科技協會（ITAA）在 2003 年就業率調查報告，IT 產業在 2002 年跌到谷底，而美國 IT 工作機會於 2003 年達到 1030 萬，比去年增加了 4.2%。勞工局統計預測從 2002 年到 2012 年 IT 工作機會將會增加更多。軟體工程師的數量會增加 179,000, 而管理顧問應該會增加 176,000。IT 人才仍然很受歡迎。系統分析者的主要角色是研究問題以及組織的需要，決定什麼樣的人、方式和資訊技術可以做最好的結合以改善組織的效能。系統分析者幫助系統使用者與其他商業管理者，制定出更新穎或更高品質的資訊服務需求。系統分析者是系統開發過程中最重要的關鍵。一個成功的系統分析者，你必須發展四種技能：分析、技術、管理和人際關係。具有分析能力能讓你瞭解組織和它的功用，找出問題所在並且分析和解決問題。系統思考是你可以去發展的重要分析能力，或是具有看出組織和它的資訊系統在系統上的能力。系統思考提供可以看出在資訊系統中重要的關連，與組織和組織本身四周環境裡的一個架構。具

備技術技能可以幫助你瞭解資訊技術的限制與潛在可能性。作為分析師，你必須能構想一個資訊系統，並且指導系統設計與發展與幫助使用者解決問題。你必須瞭解 C、JAVA 等程式語言，並且能在不同的作業系統上工作，例如：Windows、Linux，且能跨平台作業像是 IBM 或是 Mac。具備管理技能幫助你管理好專案、資源、風險與變動。具有人際關係的能力可以幫助你在工作上能與終端使用者有良好的互動並且和其他分析師與程式設計師溝通。作為一個系統分析者，你將扮演一個重要的角色介於使用者、程式設計師與其他系統專家三方的溝通橋樑。有效的簡報與溝通，包含能主導會議、客戶訪談與聆聽，這是分析師必須掌握的關鍵技巧。專業的分析師必須成功的兼具這四種技能，就像圖 1-8（系統分析者的徵人啟事）說明。

Simon & Taylor, Inc., a candle manufacturer, has an immediate opening for a systems analyst in its Vermont-based office.

The ideal candidate will have:

1. A bachelor's degree in management information systems or computer science.
2. Two years' experience with UNIX/LINUX.
3. Experience with C, Java, and/or other object-oriented programming languages, and with application development environments such as Visual Studio or RationalRose.
4. LAN-related skills and experience.
5. Familiarity with distribution and manufacturing concepts (allocation, replenishment, shop floor control, and production scheduling).
6. Working knowledge of project management and all phases of the systems development life cycle.
7. Strong communication skills.

We offer a competitive salary, relocation assistance, and the challenges of working in a state-of-the-art IT environment.

E-mail your resume to HR@simontaylor.com.

Simon & Taylor, Inc., is an equal opportunity employer.

圖 1-8
系統分析者的徵人啟事

讓我們舉兩個例子，看看一個系統分析者可能會面對的各種組織問題。第一種狀況，你在一個雜誌公司的資訊部門工作，這公司在保存和更新訂閱者名單的工作上有些問題，而且有些顧客會收到兩本雜誌，如果這種情況繼續下去，公司會損失金錢與流失訂閱者。為了創造更有效率的追蹤系統，現在電腦系統的使用者也和財務的管理者一樣，把他們的問題提交給你和你資訊部門裡的同事。第二種狀況，你在大學裡的資訊系統部門工作，在發生像學生的成績單寄錯地方的問題時，你被要求找出組織的問題所在。 當發展資訊系統

去處理這類的問題時，一個組織與它的系統分析者需有數個選擇：他們能找可靠的 IT 服務商，例如 Accenture 或 EDS 等顧問公司來為他們發展系統；他們也可以買現成的系統，使用一個類似 SAP 公司的廣泛型企業系統；他們可以使用開放原始碼軟體；或是找公司員工來發展系統。公司可以決定系統要委外還是自己開發。

1.4.2 委外

委外（Outsourcing）
將公司的一些或全部的資訊系統應用和運作委託外面的公司負責。

如果其他組織為你的組織發展或管理電腦應用程式，這就是所謂的**委外**。一個極端的例子就是將你的應用程式的發展和管理全部委託在一個公司的電腦－你只提供輸入和接受輸出。常見的例子就是公司為客戶管理薪資帳單應用程式，如此一來，客戶不需要在公司內部發展獨立的薪資系統，相反地他們只單純地提供薪資資訊給公司，而公司將回覆員工完整的薪資、薪資帳單的會計報告、稅、和其他的報表。對於許多的組織來說，以這種委外的方式處理薪資帳單是一個有成本效益的作業方式。在另一個委外工作分配的例子，你雇用一個公司管理你公司電腦中的應用軟體，在某些情況，組織採用這樣的安排可以將免去部分或全部的資訊系統單位，甚至可以裁撤所有的資訊部門的員工。然而大多數的狀況是公司在開始運行組織的資訊處理時，往往會雇用資訊系統單位的人員。

網路搜尋
委外變得非常普遍，為組織開發與運作他們的資訊系統，很多人相信它將持續成長。請造訪http://www.prenhall.com/valacich取得所有相關主題。

為什麼組織要將其資訊系統的作業委外呢？就像我們在薪資帳單的例子中所看到的，委外蠻符合成本效益原則，如果一個公司專門為其他公司管理薪資帳單，它能夠只以一個穩定的電腦應用程式為許多組織服務，進而降低成本，產生規模經濟的財務槓桿，但是為什麼組織要將它整個資訊處理單元分配並交由外面的公司管理它的電腦應用程式？一個理由也許是為了克服組織在它資訊系統單元所面對到的作業問題，例如，密西根的 Grand Rapids 市，為了更有效地管理其電腦中心的員工，三十年前即將其電腦方面的工作交由外部的公司管理，由於聯邦契約與市民服務（Union contracts and civil service）的限制使得市政單位難以解雇它的員工，所以該城市引進設備管理系統來管理它電腦方面的工作，同時並解雇有問題的員工。全面委外的另一個理由是組織的管理部門也許認為其核心的工作並非管理資訊系統單位，此外將所有資訊系統單位的工作交由更有經驗、電腦導向的公司也許獲得更有效率的運算，例如，柯達在

1980 末期就決定它不是一個電腦應用的企業，而把主機的管理交由 IBM，並將個人電腦交由 Businessland 管理。

委外是一個龐大的交易市場。一些組織委外 IT 的開發和許多它們 IT 的工作，價值值數十億美金。 例如：美國國防部在 2003 年簽署了 IT 委外合約價值 830 億美金，這是一筆龐大的金額。全部的委外合約包含的服務－軍隊、海軍、空軍，這份國防部合約的總額由許多供應商共同簽署，而各自的委外供應商也簽署一份合約為他們服務。IBM 與 EDS 是其中最大且最知名的兩家委外公司，這兩家公司有許多來自不同企業的委外合約。IBM 的客戶包含美國航空，其合約價值 40 億美金，和 Qwest 通訊的 20 億美金合約。然而，委外不是沒有潛在風險的。一家大型的金融企業 - 日本摩根，取消了與 IBM 一筆 50 億的合約，和 EDS 宣布它與美國海軍 70 億美金的委外合約發生問題。

委外是系統分析者必需注意到的一個替代方案。假如你是系統分析者，當為系統產生替代系統發展方案時，你應該查詢在該領域提供委外服務的組織，也許正好有這樣的組織已經發展並運作應用軟體，而他們所提供的非常接近你的使用者的需求。也許委外是你替代系統的替代方案之一，在你考慮委外的方法前，你應該先了解你的系統需求，這樣你才能謹慎地評估委外服務的提供者所提供的服務是否滿足你的需求。無論如何，你所應決定的並不是考慮委外，你需要考慮的是替代系統中的某些軟體元件是否應該向外購買，而非自行發展。

1.4.3　軟體的來源

我們將製造軟體的組織分為五個主要的類別：

1. 資訊技術服務公司
2. 套裝軟體供應商
3. 企業軟體解決方案的製造商
4. 開放原始碼軟體
5. 企業內部自行開發

資訊技術服務公司

如果企業需要資訊系統但又沒有技術或是員工發展內部系統，而現成系統又無法使用，企業很有可能會諮詢資訊技術服務公司。IT 服務公司幫助企業發展資訊系統提供內部使用，他們會為客戶開發、建置和運作應用程式，或者提供其它服務。表 1-1 是全球前十名軟體公司名單，有六家是此服務的專業，包含顧客系統開發。這些公司雇用有資訊系統開發背景的員工，他們的顧問在特定的業務領域也都很專業。例如：在銀行工作的顧問不但瞭解銀行與金融界的制度也懂資訊系統，所以顧問與公司一樣採用相同的方法、技術或工具協助開發系統。

網路搜尋
頂尖的技術與軟體企業的名單排名是很容易取得，請造訪http://www.prenhall.com/valacich取得所有相關主題。

也許會你會感到很驚訝，看到 IBM 被列為全球最知名的軟體供應商。即使你已閱讀有關於 IBM 是全球委外市場的領導者的個案，你也許還是會認為 IBM 是一家主要提供硬體服務的公司，。在 2002 年 IBM 併購了 IT 顧問部門 PricewaterhouseCoopers，將服務與顧問整合進來，帶領其它的 IT 服務公司，包含傳統的顧問公司像是 EDS、Lockheed Martin、Computer Sciences Corp. 和 Accenture，並且這份名單包含 HP 和其它以前專做硬體轉型成 IT 服務的公司。

表 1-1
2004年全球前十大軟體公司

排名	公司	2003 軟體/服務營收（百萬美元）	軟體商業領域
1	IBM	$56,946	IT服務 / 資源 / 中介軟體 / 應用程式伺服器 / 網頁伺服器
2	微軟（Microsoft）	$32,190	作業系統
3	EDS	$20,000	IT服務 / 資源
4	Lockheed Martin	$15,275	IT服務 / 資源
5	Computer Sciences Corp.	$13,846	IT服務 / 資源
6	Accenture	$13,397	IT服務 / 資源
7	HP	$10,165	IT服務 / 資源
8	甲骨文（Oracle）	$9,475	資料庫 / ERP
9	Hitachi	$9,345	其他
10	SAP America	$18,779	ERP

套裝軟體製造商

自從 1960 中期，軟體產業便開始以驚人的速度成長，就像軟體雜誌所估計的，現在世界上有些最大的電腦公司就是專門生產軟體的公司，見表 7-1，像 Andersen Consulting（第二十名）這類諮詢公司也列在前五十大套裝軟體製造商之列。 軟體公司發展那些所謂的商業套件系統，微軟的 Project、Intuit 的 Quicken、QuickPay、和 QuickBooks 就是此種軟體常見的例子。套裝軟體發展產業供應許多市場區隔，它所提供的軟體範圍從像日記簿那樣一般而普遍套裝軟體，到像輔助管理日間托兒中心這類範圍狹小而特定的套裝軟體。軟體公司發展許多從微電腦到大型主機電腦各種不同電腦平台上運作的軟體，這類軟體公司的規模從幾個人到數千個員工。在最初軟體設計完成和系統初期版本建置後，軟體公司會與系統使用者商議，接下來該系統會在實際的組織中進行測試，以決定有無任何問題或有無任何可改進的地方，直到測試完成，系統便會銷售給大眾。

有些商業軟體系統不能被調整成完全符合一個特定組織特殊、個別的需求。這類的應用系統有時被稱為“turnkey system”，只有在相當數量的使用者要求某種特別的改變時，“turnkey system”的生產者才會對軟體進行改變，其他的商業應用軟體則能夠被生產者或使用者調整或擴充以更加符合組織的需求，即使許多組織有著類似的功能，但是沒有兩個組織會以相同的方式做同樣一件事，“turnkey system”在某一程度的表現上也許夠好，但是它永遠無法完美地符合特定組織從事商業行為的方式，經過合理的評估，商業軟體最多能符合一個組織 70% 需求，因此，即使在最好的情況下，軟體系統的 30% 無法符合組織的特別需求。

企業軟體解決方案

就像之前所提到的，越來越多的組織選擇完整的軟體解決方案來支援它們的作業與企業流程，這些軟體解決方案稱做**“企業解決方案（enterprise solutions）”**或**“企業資源規劃（enterprise resource planning, ERP）”**。這類的 ERP 軟體解決方案包含一系列已經整合的模組，每一個模組分別支援一個傳統的企業功能，例如會計、鋪貨（通路）、製造、和人力資源，模組與傳統方法

企業解決方案軟體（Enterprise solutions software）/企業資源規劃（Enterprise resource planning, ERP）系統
一個整合傳統企業功能至一系列模組的系統，以便讓一個交易作業發生在一個資訊系統而不是好幾個分散的系統。

的不同處在於模組是經過整合的，著重於企業流程而非企業功能領域，例如，一系列的模組將支援整個訂單處理流程，從收到訂單到調整存貨、到運貨、到寄出訂單、到售後服務。傳統的方法則在不同的企業功能領域使用不同的系統，例如，帳務系統歸於會計系統，存貨系統則屬於倉庫系統。使用企業軟體解決方案，一個公司能整合在聯合資訊系統中企業流程的所有部分，單一交易的每一環節都能緊密地發生在單一的資訊系統中，而非散亂在強調企業功能領域的一系列的系統中。

網路搜尋
對於企業資源規劃（ERP）系統有興趣，這裡有可用的情報。請造訪http://www.prenhall.com/valacich取得所有相關主題。

採用企業解決方案的優點在於它將各方面的企業流程集中儲存，模組的使用更增加了應用上的彈性，集中儲存確保了資料的一致性與正確性，並且只需要較少的維護工作。模組非常具有彈性，因為一旦系統有需要時，只要將額外的模組加入基本系統就可以了，而新增的模組即可馬上與現有的系統整合。

企業解決方案軟體同樣也有缺點，因為這類系統非常複雜，所以需要很長的時間才能導入完成，一般組織並不需要自備專家（expertise in－house）來操作系統，因為那是非常昂貴的，所以它們必須依賴諮詢者或軟體賣方的員工。在某些情況，組織必須改變它們商業活動的方式，才能享受到從企業解決方案的轉換中所獲得的利益。

這有許多主要企業解決方案軟體的賣供應商，其中最廣為人知的大概是 SAP AG，一個以它的旗艦產品 R/3 聞名的德國公司，SAP 代表資料處理中的系統、應用和產品（Systems, Applications, and Products in Data Processing），SAP AG 建立於 1972 年，但是它的蓬勃發展是發生在 1992 年。到了 1998 年，SAP AG 已是全世界第六大的軟體供應商（見表 7－2）。另一個企業解決方案軟體的供應商是 PeopleSoft, Inc.，一個建立於 1987 年的美國公司，PeopleSoft 公司開始是著重於人力資源管理的企業解決方案，但是現在已經擴充到涵蓋財務、物料管理、經銷和製造等方面。第三個供應商是 Baan 公司，一個建立於的 1978 荷蘭公司，Baan 的應用軟體包製造、財務、經銷和運輸、和服務，其他兩個主要的企業解決方案的供應商是 Oracle 和 J.D.Edwards 這兩家美國本土公司。在被 Oracle 併購之前，PeopleSoft 於 2003 年併購了另一個的 ERP 供

應商 J.D. Edwards，讓它的公司更強大。全球製造業的 ERP 市場營收預估從 2002 至 2007 年將由九十億美金成長到一百二十億美金（arcweb.com, 2003），由於 ERP 系統市場的頂端已呈現飽和狀態，所以有許多 ERP 供應商開始看到中小型企業的成長空間。

開放原始碼軟體

開放原始碼軟體不同於目前你所看到其它類型的軟體。開放原始碼軟體的不同在於它是可以自由的使用，不是一個完整的產品，它只有原始碼。它很特別是因為它是由一群有共同興趣的社群所開發。開放原始碼軟體所做的與商業軟體功能上大同小異，例如：作業系統、e-mail、資料庫系統、網路瀏覽器等。一些知名與受歡迎的開放原始碼軟體像是 Linux（作業系統）、mySQL（資料庫系統）和 Firefox（網路瀏覽器）。開放原始碼也常被用於當作軟體的元件與物件。開放原始碼是由社群的人們所開發與維護，有時後這些社群規模是非常大的。開發者經常使用公開的網路資源，像是 SourceForge.net 讓他們可以在此活動。在 2005 年二月，SourceForge.net 主機上已經有 95,000 個專案和一百萬以上的註冊會員。無庸置疑的，假如沒有享受在網路上存取資料與組織開發的活動，這個開放原始碼運動將不會成功。

如果軟體是免費的，你會想知道他如何取得資金來開發開放原始碼軟體。企業與個人可以用開放原始碼賺取資金，主要有兩個方式：（1）提供維護和其它的服務（2）提供一個免費的軟體版本和販售一個完整功能的版本。一些開放原始碼的解決方案衝擊到許多軟體產業，Linux 就是個例子，在伺服器軟體市場非常的成功，在市場佔有率估計有 24%，在 2007 年將成長到 33%。在桌上型作業系統，Linux 目前有 3% 的市場佔有率，在 2007 年會倍數成長到 6%。現在這個市場佔有會轉移到市值 110 億美金的 Linux，在 2008 年潛在的市值將高達 357 億美金。其它的開放原始碼軟體，像是 mySQL 也相當的成功。開放原始碼瓜分了現有的軟體產業而且將會繼續成長。

企業內部自行發展

我們已經談過關於四種由外部組織取得軟體的來源，但是由企業內部自行開發也是一項選擇，當然，企業內部自行發展並不需要負擔整個系統所有軟體的發展，較常見的是混合式的解決方案，包括部分購買和部分自行發展軟體元件，在表 1-2 中比較了這五種不同的軟體來源。

表 1-2
五種不同軟體來源的比較

表 1-2 缺譯

1.5 資訊系統與系統發展的類型

組織中具有廣泛領域的人才與興趣，可以涵蓋多種不同類型的資訊系統，以滿足組織中所有資訊系統的需要。目前為止，我們只是對資訊系統做一般性的概述，但是實際上資訊系統分成數個不同的類型或種類。類型與類型之間的區分，是基於系統的功能或是建構系統時所用的技術。作為一個系統分析者，你部份的工作重點，是決定哪一種系統的類型最能找出組織問題或機會。另外，不同的系統種類在發展上會需要不同的方法、技巧以及工具。 當系統分析者在團隊中工作時，至少會接觸到三類的資訊系統：

- 交易處理系統
- 管理資訊系統
- 決策支援系統（為個人、團體以及執行部門）

這些系統類型表示於圖 1-9 中，而每一種主要系統的類型將在後面的章節中做更詳細的說明，圖 1-9 顯示他們之間的一些差異。在圖中的交易處理系統（Transaction Processing Systems），你可以看到主要是針對交易資料的擷取，然後將這些資料傳送到所有業務的電腦化資料庫中。在男性的圖像對應的是現金記錄，而女性的圖像是取得訂單，兩者都表示交易資料的取得，從圖像到電腦的箭頭表示為了儲存而將資料移到資料庫中。圖 1-9 說明管理資訊系統（Management Information Systems）的圖案，是顯示一個管理者利用交易資料產生一份關於上個月銷售情況的報告。管理資訊系統是被設計用來把交易資料處理成標準的報告。圖 1-9 中的另一個圖案表示決策支援系統（Decision Support Systems）。決策支援系統幫助管理者藉由不同的資料分析方法來做出決定。管理者能夠改變他們的資料，就像改變利率，然後觀察那些變化如何影響他們所管理的商業部份。圖 1-9 的圖案中，管理者嘗試決定什麼程度才會把向下的趨勢轉成向上的趨勢。下列的章節簡短地說明系統分析與設計方法在四種主要類型系統之間有何種差異。

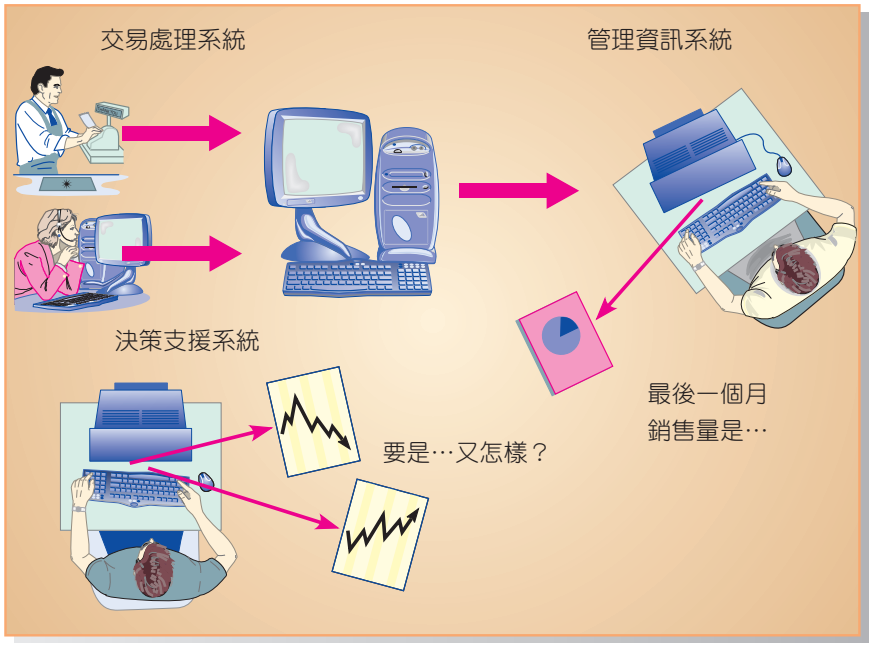


圖 1-9
三種資訊系統：TPS、MIS和DSS

1.5.1 交易處理系統

交易處理系統（Transaction Processing Systems , TPS）自動地掌控商業活動或業務的資料。例如，一家銀行的 TPS 能從顧客帳戶獲得提款和存款的資訊。每一筆業務的資料被取得後，業務會被裁定是接受或被拒絕，然後有效的交易會被儲存。報告可能立即的被產生出來總結業務的歷程，而為了掌握所有的商業活動，業務可能從這個過程被移動到下一個過程。

TPS 的分析與設計需要集中注意力在公司目前處理業務的程序。組織是如何的追蹤、獲得、處理以及輸出資料？TPS 發展的目的就是改進業務的處理，藉由提昇速度、使用較少的人員、提高效率與準確性、整合其它組織的資訊系統、或者提供先前無法取得的資訊等方式來達到這個目的。

1.5.2 管理資訊系統

管理資訊系統（Management Information Systems, MIS）是一個以電腦為基礎的系統，它透過 TPS 取得可用的原始資料，並且轉換這些資料成為有意義的形式。舉例來說，鑒於業務管理系統需要記錄銷售量，管理資訊系統可以精確的知道哪一個項目的銷售情形是緩慢的，而哪一個項目的銷售情形是快速的。MIS 系統因而可指示製造部門何時該產生何種產品。發展一個 MIS 需要很了解什麼樣的資訊是管理者所需要的，以及管理者如何在他們的工作上使用資訊。有些時候，管理者自己本身也不清楚什麼是他們所需要的，或是他們該如何使用資訊。因此，分析師也必須具有良好的商業認知，並了解為 MIS 提供資料的交易處理系統。

管理資訊系統經常從數個交易處理系統（例如，客戶訂單的處理、原料的購買以及員工計時工資）要求資料。因此，MIS 的發展可以從資料的取向得到好處，也就是從 TPS 取得的資料中區分出可作為組織資源的資料。因為從各種主題領域中汲取資料是相當重要的能力，開發綜合與準確的資料模型是建構管理資訊系統的必要工作。

1.5.3　決策支援系統

決策支援系統（Decision Support Systems, DSS）是被設計用來幫助決策者做決定。一旦 MIS 產生報告後，決策支援系統提供一個互動的環境，讓決策者可以很快的運用商業運作的資料與模型。DSS 包含三個部分：第一部份是由資料庫組成（可能從 TPS 與 MIS 取得）。第二部分由商業處理的數學與圖形模型所組成。第三部份是由使用者介面（或對話模組）組成，它提供給決策者與 DSS 溝通的一個方式。對於過去未曾發生或未來可能的狀況，DSS 能同時使用歷史資料以及判斷（或稱為“What if”的分析）的方式來運作。執行資訊系統（executive information system, EIS）這樣的 DSS，可以讓資深的管理部門在高度聚集的狀況下探索資料，並選擇性的深入特定領域取得所需之進一步資訊。DSS 的特性是具有較低的結構性與可預測性，DSS 軟體可用來支援某些決策活動（從發現的問題到選擇行動流程）。 DSS 的系統分析與設計通常集中於 DSS 的三個主要元件內：資料庫、模型基礎、和使用者對話。就如同在 MIS 中，資料的取向常常被用於瞭解使用者的需求。系統分析與設計的計畫將小心制定出不同資料彼此間關係上的數學規則，這些關係被用來預測未來的資料或找出決定問題的最佳解決方式。決策邏輯必須小心地理解與記錄。同樣地，一個決策者通常會與決策支援系統互動，因此易用的使用者對話框與螢幕畫面的設計是相當重要的。

表 1-3
不同IS類型的系統發展

表 1-3 缺譯

1.5.4 資訊系統：概觀

許多你所建立或維護的資訊系統往往會包含此三種主要類型資訊系統的每一種形態。一個系統分析者將會使用特定的方法、技巧與工具來聯繫此三種資訊統類型。表 1-3 概述每一種類型的特色與發展方法。 我們會注意資訊系統是在哪裡被發展以及被誰維護，我們也能看到存在於組織內部的三種類型的資訊系統。現在，我們轉向資訊系統的發展過程－系統發展的生命週期。

系統發展方法（Systems development methodology）
組織使用一套標準的程序分析、設計、車實作與維護資訊系統

系統發展生命週期（Systems development life cycle, SDLC）
系統發展每個階段的階段

發展資訊系統與系統發展的生命週期　組織使用一套標準的步驟，稱作**系統發展方法**，來發展與支持他們的資訊系統。就像許多的處理過程，資訊系統的發展通常伴隨著生命週期。舉例來說，像是 Nike 運動鞋或是 Honda 汽車這類的商業產品都有生命週期：被建立、測試與引進市場，它的銷售量增加、到達顛峰然後下降，最後，這些產品將從市場淘汰並被其它產品取代。**系統發展生命週期**（SDLC）是在許多組織中發展系統常見的方法學。它標註出資訊系統發展的階段與步驟：某人對資訊系統以及它該怎麼運作的想法，在仔細的評估此系統將如何支援組織目前處理的工作後，使用系統的組織會決定是否交出資源來獲得該資訊系統。專家為設計新的系統開發了一些策略，然後這些策略不是被建構就是被購買。一旦完成之後，系統將被安裝在組織內部，然後施予合適的訓練，使用者就可以將新的系統融入到他們的日常工作中。每一個組織使用些微不同的生命週期模型去塑造出這些步驟，從三個到幾乎二十個階段不等。在這本書中，我們著重四個 SDLC 的階段：（1）規劃與選擇、（2）分析、（3）設計與（4）實作與操作。

雖然任何生命週期起初看起來是一個連續而有順序的階段集合，但事實上並非如此。圖 1-10 強調 SDLC 中的四個階段：（1）規劃與選擇（planning and selection）、（2）分析（analysis）、（3）設計（design）與（4）實作與操作（implementation and operation）。這些特定的階段與順序是依計畫的需要而定的。舉例來說，在任何所給定的 SDLC 的階段，如果需要的話，計畫可以返回較早的階段。同樣地，如果一個商業的產品不能在它引進之後很好地完成，它可能會暫時的被市場排除並在重新引進之前被改善。在系統發展生命週期裡，它也可能在這一階段與另一階段平行的完成一些活動。有些時候生命週期是反覆的，也就是說階段會一直被

重複直到找到一個可以接受的系統。像這樣在不同層次的細節上不斷的在某些階段做循環，有些系統的分析師把這樣的生命週期發展視為一個螺線，如圖 1-11 所示。圖 1-11 中生命週期圓形圖示的性質說明一個系統的使用壽命如何結束以至於引導另一個計畫的開始，這些計畫將完全的取代已存在的系統。不管如何想像，使用於組織內的系統發展生命週期是一組井然有序的活動集合，而這些活動是為了每一個發展計畫而運作和規劃的。系統分析者所具備的技能，要能應用到所有生命週期的模型中。

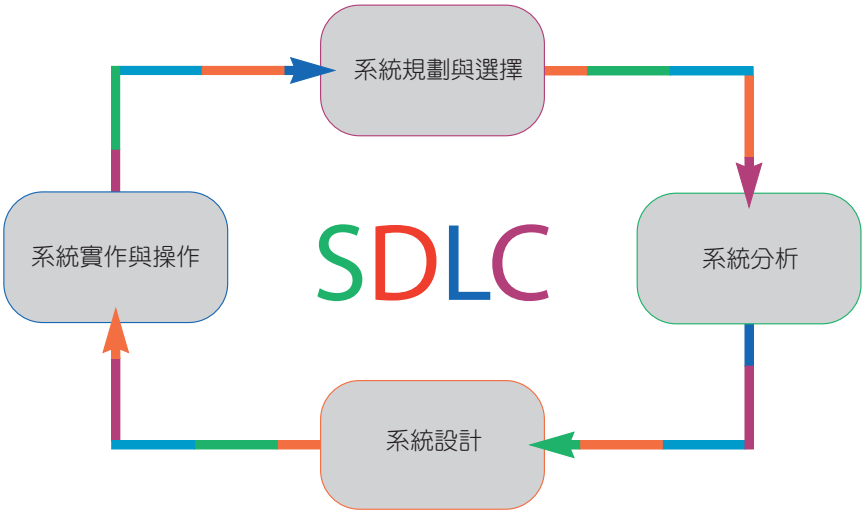


圖 1-10
系統發展生命週期（SDLC）

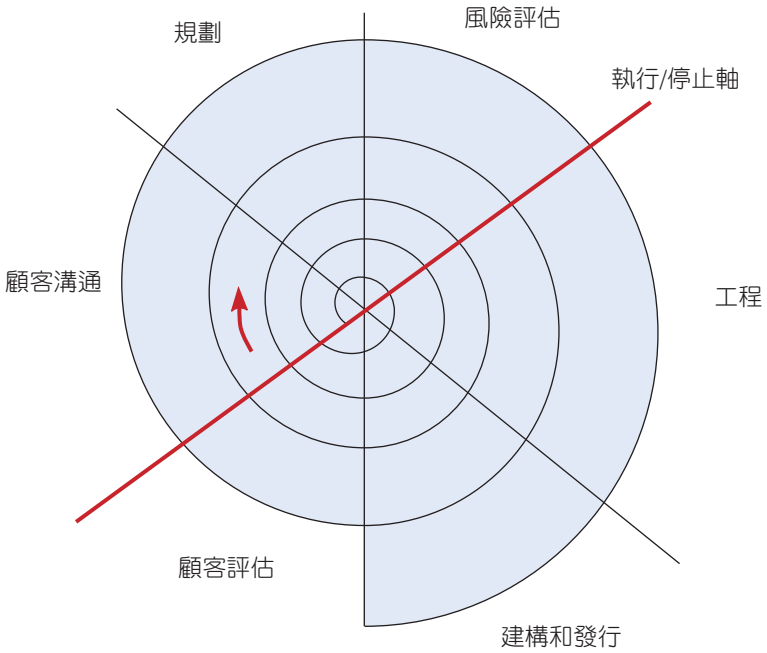


圖 1-11
SDLC發展模式

每一個較具規模的公司如 Wal-Mart 或 Kmart 與每一個訂製軟體的生產者如 SAP 會有屬於自己特有的、詳細的生命週期或系統發展方法。即使某個特定的方法看起來不像一個週期，許多 SDLC 的階段、技巧與工具還是被使用。這本書依循一個總括性的 SDLC 模型，如圖 1-10 所示，我們使用這個 SDLC 作為一個範例方法，以及一個系統分析與設計的思考方向，你幾乎可以將這個方法應用到任何一個生命週期上。我們在全書所敘述的 SDLC 觀念，可以讓你清楚的明瞭每一個階段會有特殊的結果，並傳送重要資訊給其它的階段。在每一個階段的結尾（和有些時候在階段的中間步驟），系統發展的計畫會到達一個里程碑，然後會產生一些成果，它們通常會被計畫團體之外的一群人重新探討，這些人包括管理者與執行者。

1.5.5 階段一：系統規劃與選擇

系統規劃與選擇（Systems planning and selection）
SDLC的第一個階段，在此階段中，組織的資訊系統需要進行完整的分析與規劃，定義潛在的資訊系統專案，並考量系統發展的專案是否要繼續進行。

SDLC 的第一階段是**系統規劃與選擇**，有兩個主要的活動。第一，有人確定需要新的或增強的系統。組織的資訊需求要被考核，而且能解決需求問題的計畫也要被確立。 組織資訊系統的需求可能源自於：

- 要求處理目前程序中的問題
- 要求完成附加的工作
- 了解資訊技術可能在現有的機會中帶來更大的利基

系統分析者為了資訊系統的發展，會把事情優先順序排定好，並且轉換需求成為書面計畫，包括一個為了發展新系統的時間表。新系統的需求是來自於使用者對新的或增強系統功能的需要。在系統規劃與選擇的階段，組織要決定是否該將資源投入資訊系統的發展或增強上。可行性研究（feasibility study）會在 SDLC 的第二個階段之前被導入，以決定出系統對經濟與組織的衝擊。

系統規劃與選擇的第二個工作是調查並決定系統的範圍。此時系統分析者團隊會為了團隊要依循的計畫而製作一個特定的計畫，為此計畫案製訂標準化的 SDLC，並詳細說明它執行時所需要的時間與資源。組織的資訊系統部門能否發展出一個具有解決問題或利用機會的系統，以及決定發展一個系統的代價是否高於可能的利益，這

些可能性決定了一個計畫是否能正式成立。而對進行後續階段所做的計畫，其管理工作通常是由計畫領導人與其他的團隊組員提出總結性的報告。

1.5.6　階段二：系統分析

系統發展生命週期的第二個階段是**系統分析**。在這個階段，系統分析者徹底的學習組織現有的程序以及運用在工作上的資訊系統，如一般的算帳、運輸、訂貨記錄、機器的時間表與薪資處理。「分析」包含了數個子階段。第一個階段專注於決定系統的需求，在這個階段中，你和其他分析師與使用者一起工作，以決定使用者想要從何種系統開始。這個子階段包含任何現行系統、手冊與電腦化的徹底研究，這個計畫中的某一部份可能被替換或增強。接下來，你要根據它們彼此的相互關係，排除任何重複的部分，研究它們的需求與結構。第三，你產生可替代的初始設計以符合需求。你在比較這些替代方案後，找出在發展過程中費用、人工與技術層次上的代價都是組織願意付出的，能夠符合需求的最佳方案。分析階段的輸出結果，是由分析團隊所推薦之替代解決方案的陳述。一旦這個推薦被組織所接受，你可以獲得準備計畫所需的任何硬體以及所需要的系統軟體去建立或產生建議的系統。

系統分析
（Systems analysis）
SDLC的一個階段，在此階段會探討目前的系統，並提出可替代的系統。

1.5.7　階段三：系統設計

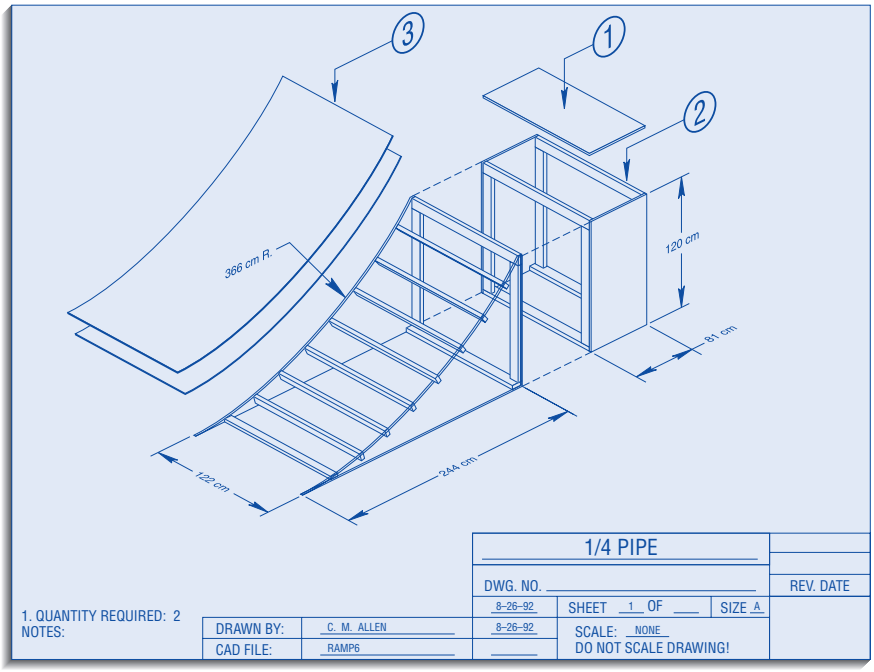
SDLC 的第三個階段稱為**系統設計**。在系統設計階段，分析師將所推薦之替代解決方案的陳述，轉換成數位邏輯，然後再轉換成實體的系統規格。你必須設計系統的所有風貌，從輸入與輸出的螢幕畫面到報告、資料庫以及計算過程。

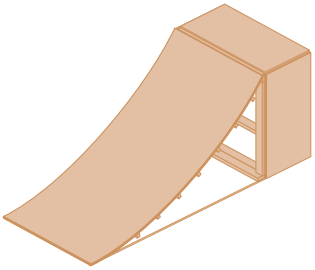
邏輯設計不會受限於任何硬體與系統軟體的平台。理論上，你所設計的系統要能在任何的硬體或系統軟體上被完成。邏輯設計集中在系統的商業觀點；也就是說，系統將如何影響組織內部的機能單元。圖 1-12 顯示一產品的邏輯設計以及它的實體設計。你可以從比較中看到，當產品從邏輯的模型移到實際成果的過程中，做出了許多特殊的決定，這情況非常類似資訊系統的設計。

系統設計
（Systems design）
SDLC的一個階段，依據系統分析先描述獨立於不同電腦平台的系統設計（邏輯設計），然後再用特定的程式語言或系統建構技術轉成實際系統的詳細設計（實體設計）

圖 1-12

缺





在實體設計上，你轉換邏輯的設計成為實體的、專門的與詳細的說明。舉例來說，你必須轉換系統中的原始設計圖、流程圖與資料處理過程到一個有結構的系統設計，這樣可以將系統向下縮減成越來越小的單元，以轉換成程式語言所能撰寫的指令。你設計系統的許多部分去執行實體所需的操作，以促進資料的獲得、處理過程以及資訊輸出。在實體設計期間，分析團隊決定哪種程式語言的電腦指令將被撰寫，資料將使用哪種資料庫系統與檔案結構，以及哪種硬體平台、作業系統與網路環境，這些都將執行在系統之下。這些決定終結了始於分析階段後期硬體與軟體計畫。現在你可以獲得任何新的且不存在於組織中的技術。設計階段的最終產品是實體的系統規格書，以圖形或書面報告的形式表現，然後丟給程式設計人員或其他的系統建構者建造。

1.5.8　階段四：系統實作與操作

SDLC 的最終階段是一個兩步驟的處理過程：**系統實作與操作**。在系統實作與操作階段，你轉換系統規格到一個可作用的系統，這系統將被測試然後進入使用的階段。實作包括程式碼的撰寫、測試與安裝。在程式碼的撰寫期間，程式設計人員撰寫程式以組成系統。在測試期間，為了找到並更正錯誤，程式設計人員與系統分析者測試個別的程式以及全部的系統。在安裝期間，新的系統將變成組織中日常活動的一部份。應用軟體被安裝或被下載到硬體上，然後使用者將被導入新的系統並接受訓練。測試與安裝的計畫的開始時間為要和計畫與選擇階段一樣早，因為兩者都需要大量的分析，早早開始才能發展出完全正確的方法。

系統實作與操作（Systems implementation and operation）
SDLC的最後階段，在此階段包括資訊系統程式的撰寫、測試與安裝，並系統化的維修與改進資訊系統。

系統實作的行動也包括最初使用者的支援，譬如文件的完成、訓練課程、與即時的使用者輔助。文件與訓練課程在實作期間被完成，文件的產生遍及生命週期，而訓練（與教育）的則是從一個計畫開端就開始進行。只要系統存在，系統實作可能會持續，因為現在進行的使用者支援也是系統的一部份。儘管分析師、管理者以及程式設計人員都是最好的，但是安裝過程並非全都是個簡單過程。許多良好設計的系統也曾因為安裝過程的缺陷而失敗。即使是良好設計的系統，如果實作方面沒有被好好的管理，也可能會有失敗的下場。因為系統管理的實作通常是由計畫團隊來完成，我們會在這本書中強調實作的問題。

SDLC 第四個階段的第二部分是操作。當一個系統在組織內被操作的時候，使用者有時會發現系統運作上的問題，並且經常會想要改善。在操作階段，程式設計人員會進行一些改變，以符合那些使用者的要求，並且修改系統以反映出商業條件的變化。這些改變需要維持系統的運作與使用。

在操作階段，要為增強系統投入多少的時間與精力，取決於生命週期前期階段工作的效能。然而，當維持運作的費用遭到刪除時，或當一個組織的需求大幅度改變時，一個資訊系統在中途就夭折將是無法避免的命運。這些問題的發生意味著是開始設計替換系統的時候了，應該結束這個循環而重新開始另一個生命週期。

表 1-4
SDLC 各階段的產出

表 1-4 缺譯

SDLC 是一個高度連結的階段集合，這些階段會將它們的成果提供給在下一個階段的活動。基於上面的描述，表 1-4 總結出每一個階段的輸出或成果，在 SDLC 階段的後續章節中將討論每一個階段的成果以及它們如何被發展。

在整個系統發展的生命週期中，必須嚴謹的規劃與管理系統發展計畫。系統專案規模愈大，對計畫管理的需求也會愈大。數種計畫管理的技術在最近二十幾年被發展出來，並且有許多技術透過自動化的方式被改善。第二章包括更詳細專案規劃與管理技術的說明。

1.6 發展的方法

雛型法（prototyping）、電腦輔助軟體工程（CASE）、聯合應用程式設計（joint application design, JAD）、快速應用程式發展（rapid application, RAD）、參與設計（participatory design, PD），以及敏捷方法論，這些方法能有效率且能改善系統分析與設計的處理過程。

1.6.1 雛型法

雛型法（Prototyping）
為欲發展的系統設計一個精簡的版本

為欲發展的系統而設計一個精簡但可運作的版本，這叫做**雛型法**。雛型可以被電腦輔助軟體工程（CASE）工具所發展，這是一個使系統發展生命週期步驟自動化的軟體產品。由於螢幕畫面與報表

以及系統其它介面等設計上的支援，CASE 工具使得雛型法更加容易，更具想像空間。CASE 工具也提供許多的圖示技巧，例如資料流程圖與實體關係圖。

圖 1-13 說明了雛型法。分析師與使用者為決定系統的基本需求而一同工作，然後分析師很快的建立一個雛型。當這個雛型完成時，使用者會跟著這個雛型一起工作，並且告訴分析師什麼是符合他們期望的，什麼是不符合他們期望的。分析師使用這些回饋的意見去改善雛型，並且將更新的版本帶給使用者。這個反覆的處理過程會持續到使用者滿意為止。雛型法技術的主要優點是（1）它讓使用者參與分析與設計；（2）它是實際去取得需求，而不是口頭的、抽象的、流於形式的。除了可單獨使用外，雛型法也能用於擴大到 SDLC 上。舉例來說，在分析師進行了解使用者需求的分析步驟時，最終系統的雛型可能早已經被發展出來了，之後最終系統以雛型的規格書為基礎被開發。我們將在第四章詳細的討論雛型法以及在第八章使用多樣的雛型法工具來說明系統輸出的設計。

網路搜尋
對於企業資源規劃（ERP）系統有興趣，這裡有可用的情報。請造訪http://www.prenhall.com/valacich取得所有相關主題。

雛型法是一個支援快速應用發展（rapid application development, RAD）的重要工具。任何 RAD 方法的基本原則是：直到使用者的需求明確以後，才產生詳細的系統設計文件。雛型就像是需求的工作描述，而 RAD 會儘快取得使用者對介面與發展關鍵系統效能的接受度。RAD 廣泛的被顧問公司採用，如 Cambridge Technology Partners。它也被公司採用做為內部的方法，像 Hughes Electronics Corporation。RAD 會為了能獲得快速建立可運作系統中的人力效能，而犧牲了電腦效能。另一方面，對於可能導致系統發展方面問題的系統發展原則，RAD 的方法可以忽略這些原則。附錄 B 提出詳細的 RAD。

1.6.2　電腦輔助軟體工程（CASE）工具

電腦輔助軟體工程（CASE）由系統分析者使用自動化的軟體工具來開發資訊系統。使用這些工具自動化或支援系統開發過程中的所有活動，目的在提高生產力和改善整體系統品質。CASE 有助於軟體開發和整個軟體生命週期程序的自動化，有時候是整合性軟體的一部份。一般而言，CASE 協助系統建造者在處理複雜的資訊系統規劃，並且幫助高品質的系統能在有限的時間與預算內完成建置。

電腦輔助軟體工程（Computeraided software engineering, CASE）
可針對部分系統發展過程提供自動化支援的軟體工具

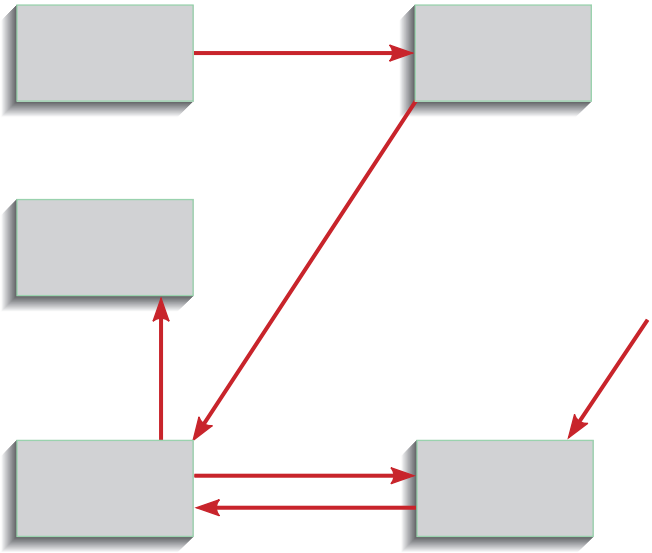


圖 1-13

缺

儲存庫（Repository）
是一個集中的資料庫，它包含所以圖形、表格和報表的定義、資料結構、資料定義、流程和邏輯，以及其它組織和系統元件的定義，它提供一些機制和架構讓資料和工具間，以及資料和資料間可以整合。

CASE 產品製造商”更加開放”他們的系統，透過使用標準化資料庫與資料轉換公用程式進行跨產品與工具的資訊分享。具整合性與標準化資料庫，稱為**儲存庫**，它是提供產品和工具整合的方法，並也已成為讓 CASE 可以處理大型且複雜的專案更容易，且可跨不同工具和產品整合資料的關鍵因素。一般 CASE 工具有：

- 圖表工具能讓系統流程、資料和控制結構以圖表來呈現。
- 電腦展示和報表產生器有助於雛形法，透過系統”看和感覺”的方法呈現給使用者，展示（或表單）和報表產生器也讓系統分析者易於確認資料需求和相關性。
- 分析工具能自動檢查在圖表、表格和報表中不完整、不一致、不正確的規格。
- 一個中央儲存庫能完整的保管規格、圖表、報表和專案管理資訊。
- 文件產生器能幫助產生使用者技術文件標準規格。
- 程式產生器能直接由設計好的文件、圖表、表格和報表自動產生程式與資料庫定義碼。

1.6.3　聯合應用程式設計

在 1970 年代末期，IBM 的系統發展人員為了集合資訊系統的需求與重新探討系統設計，而開發一個新的處理過程。這個過程被稱為**聯合應用程式設計（joint application design, JAD）**。JAD 背後的想法是建構分析工作的需求決定步驟，並預覽系統設計的工作。為了一連串由 JAD 會議的領導者所主持的密集會議，使用者、管理者以及系統發展人員會齊聚一堂。藉著將這些受到資訊系統直接影響的人齊聚一堂的機會，大家一起集思廣益，讓系統的需求與設計細節、時機和組織的資源等獲得良好的管理。團體的組員會為該資訊系統要具備何種功能而建立起共識。JAD 在某些產業中很普遍，如保險業和像 CIGNA 這樣的特定公司。我們將第四章更詳細的探討 JAD。

聯合應用設計（Joint Apllication Design, JAD）
使用者、管理者和分析師會一起工作好幾天，並在一連串密集的會議裡討論出明確的系統需求。

1.6.4　快速應用程式發展

雛形法、CASE 和 JAD 是支援**快速應用程式發展（RAD）**重要的工具。任何的 RAD 方法的基礎原理在於直到完全確認使用者需求之後，才產生詳細的系統設計文件。雛形法可以是作業的需求說明，而 RAD 是讓使用者入門並能儘快開發重要系統。RAD 被顧問公司廣泛的使用，它也被使用於企業內部像是 Hughes 電子公司。RAD 犧牲了電腦的效能來換取人們在快速建置與重建執行系統的效率。另一方面，RAD 方法會忽略重要的系統開發原則，這也許會導致系統發展產生一些問題。

快速應用程式發展（Rapid application development, RAD）
系統開發方法論在於減少時間需求來設計和實作資訊系統。

RAD 是由兩項趨勢的整合產生出來的：80 年代末，90 年代初商業行為的加速與複雜化，以及有助於系統發展與易於維護的高性能電腦化工具的出現。隨著在快速變遷，極具競爭的全球化環境中進行商業行為的變因增加，許多組織的管理階層開始質疑以兩三年的時間發展一套在完成時就過時的系統是否有意義。另一方面，CASE 工具以及雛型化軟體漸漸進入各組織，讓使用者能夠更輕易的在系統完成前看到系統的樣子與行為。在快速變化的商業環境中，為何不利用這些工具來找出更具生產力的系統發展方法？RAD 就因此誕生。

在圖 1-14 裡，在傳統 SDLC 中所經歷的階段在 RAD 中一樣會看到，但是這些階段被結合起來以形成一項更為順暢的發展技巧。在 RAD 中，為了專注於系統的功能與使用者的介面需求而犧牲較詳細的系統分析與系統效能相關問題，並縮短規劃（planning）與設計（design）階段。同時，RAD 視發展中的系統與其他系統隔絕，因此在規劃與發展中減少了配合現存標準與系統的耗時動作。RAD 通常不太著重發展週期中的序列與結構，而比較重視如何同時進行不同工作並且廣泛使用雛型化（prototyping）。絕大多數的 RAD 工作都是著重在這個部分。雖然在 RAD 裡面有可能從設計階段再回到規劃階段，但是這種情況極少發生。同樣的，雖然有可能從結束階段（cutover phase）（在這階段中系統要轉交給使用者）再回頭發展，但是 RAD 是設計來減少發展週期中這一類的過程。整個 RAD 中使用者高層次的加入以及參與意指產生出來的系統應當較由傳統技巧產生出來的更易於讓使用者社群接受（也因此更容易在結束階段實作出來）。

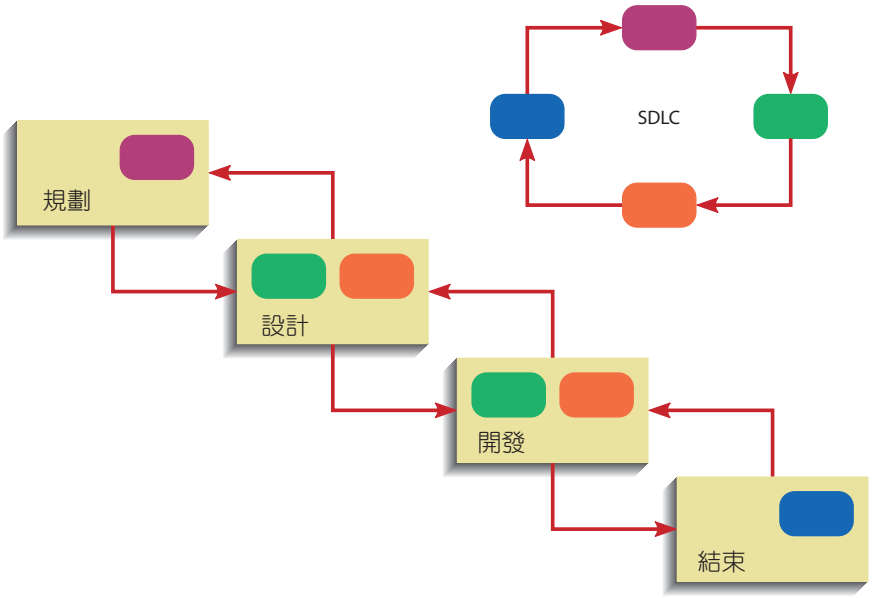


圖 1-14
相較於標準SDLC的RAD系統發展生命週期

（參考附錄 B 有更多的資訊）。這些**敏捷方法論**分為三個重要的原則：（1）重點在於適性而不是預測性的方法（2）重點在於人而不是角色（3）一個自我適應的程序。採用適性而不是預測性的方法裡提到，當程序與產品是預測性的，那麼以工程方法論進行是最好的方法。然而，軟體不是預測性，它是一個溝通的橋樑，尤其是在今天這個巨變的商業環境，所以我們需要更多的適性方法，敏捷方法論就是能根據能力快速適應。重點在於人而不是角色，人們將具有互換性。敏捷方法論是看個人具備的才能，而不是安排一個角色，每個人都有獨特的才能讓專案得以開發。最後，敏捷方法論提升成為能自我適應的軟體開發程序。當應用敏捷方法論時，系統發展的方法應該要靈活且適當的給予在特定專案上工作的特定開發團隊，而不該只用一種方法於所有開發的專案。在附錄 B 中你學得更多關於敏捷方法論的內容。

敏捷方法論（Agile method-ologies）
現今的系統開發方法重點在於適性方法、人取代了角色和一個自我適應的開發程序。

1.6.5　參與設計

發展於北歐的**參與設計（PD）**是另一套研究 SDLC 的可行方法，一個眾所皆知的挪威國營的石油公司 StatOil 就是使用這個方法。PD 較傳統的北美技術，例如在結構分析與結構設計方面，更強調使用者的角色。在有些狀況，PD 在發展的過程中會讓整個使用者社群參與，在系統需求的決定與系統設計的認可上，每一個使用者都具有相同的地位。有時，一群被推選出來的使用者團體掌控了這個程序，這些使用者代表較大的社群，如同立法機關代表了全體選民的意見。一般而言，在 PD 之下，系統分析者是為使用者工作。組織的管理部門與外部的顧問提供的是建議，而非控制。PD 可說是因北歐工作環境中勞工和管理角色而產生，因為在北歐的勞工的確比北美的較具組織，較有影響力，也較盡心投入技術上的變革。

參與設計（Participatory Design, PD）
源自於北歐的系統發展方法，著重於使用者和使用者工作的改善。

1.6.6　敏捷方法論

也許你可能會想，多年來系統分析與設計的許多方法都已經被開發，這些方法包括極致軟體製程（eXtreme Programming）、Crystal 方法、適性軟體開發、Scrum、功用導向發展方法。在 2001 年二月，這些方法的提出者在美國猶他州會面，而且根據他們不同方法的基本準則達成共識，這個共識變成發表的“敏捷宣言”文件。

重點回顧

1. 定義資訊系統的分析和設計。

系統分析和設計是複雜的組織流程是以電腦為基礎的資訊系統所開發與操作。

2. 結合系統的流程和資料的觀點，探討現代系統分析與設計的方法。

程序導向著重於什麼是系統所應該做的，而資料導向則著重於系統需要操作的資料。程序導向所提供的穩定設計比資料導向低，因為商業的程序的變化會比組織使用的資料的變化還快。使用程序導向，資料檔案是為了特定的應用而設計，而資料導向則是為了整個企業而設計資料檔案。程序導向會導致相當多不受控制的冗餘資料。

反之在資料導向之下資料的冗餘是受到控制的。資料導向與應用程式各自獨立（從電腦應用所使用的資料來分離資料）是你在這本書中學習關於系統分析和設計的架構。

3. 敘述系統分析者在資訊系統發展中的組織角色。

系統分析者在系統發展過程中扮演一個主要的組織角色。他們一方面扮演使用者間的聯絡者角色，另一方面要扮演技術人員。為了成功的達成目的，分析師需要發展四種技巧：分析上的、技術上的、管理上的，以及人與人之間的。

4. 描述三種不同的軟體來源。

可由資訊技術服務公司、套裝軟體供應商、企業解決方案的供應商、自由軟體提供者或網路系統發展資源取得應用程式（或系統），甚至也可以雇用一個組織助處理所有系統發展的需求，其稱為委外（outsourcing）。所以必須知道如何在這些軟體產品中評選的標準，這些標準包含成本、功能、供應商支援、供應商能力、彈性、文件、回應時間和安裝的容易性。此外，也必須決定是否需要新硬體和軟體，需求計畫書是收集更多關於硬體、軟體、成果和成本資訊的方法。除了硬體和軟體外，還要考量執行和組織推廣的議題，例如取得資金和管理支援。

5. 描述資訊系統的三種型態：交易處理系統、管理資訊系統和決策支援系統。

有許多不同的資訊系統被使用在組織中。這些包含交易處理系統、管理資訊系統和決策支援系統。發展的技術隨著系統的型態而變化。

6. 敘述資訊系統發展的生命週期（SDLC）。

在這本書中使用的系統發展生命週期有四個主要的階段：（1）系統規劃和選擇（2）系統分析（3）系統設計（4）系統的實作與操作。

在第一個階段，是規劃和選擇，分析師會做出詳細的系統發展計畫藍圖。在分析階段，分析師會試著解決已知的商業問題。在設計階段會建立問題的解決方案。最後，在最終的階段，系統會交予使用者並持續的執行。

7. 列出有別於系統發展生命週期的其他項目。簡短的說明電腦輔助軟體工程（CASE）工具在系統發展中的角色。

在這一章提到了四個另外的架構，分別的是雛型法、聯合應用設計（JAD）、參與設計（PD）和敏捷方法論。使用雛型法，分析師建造一個系統的工作模型。在 JAD 中，分析師和使用者相會去解決問題並設計系統。在 PD 中，強調的是使用者的溝通方面。敏捷方法論強調適性而不是預測性的方法，重視人而不是角色，並著重對處境的持續適應性。CASE 工具是指將資訊科技使用在協助系統發展的過程。它們包含圖表工具，螢幕畫面和報告的設計工具，以及其他特殊目的的工具。CASE 藉由自動的程序任務幫助程式設計師和分析師更有效的進行他們的工作。

重要名詞

這些是這一章的關鍵名詞。

1. 敏捷方法論（Agile methodologies）

2. 應用軟體（Application software）

3. 界限（Boundary）

4. 凝聚（Cohesion）

5. 元件（Component）

6. 電腦輔助軟體工程（computer-aided software engineering, CASE）

7. 限制（Constraints）

8. 耦合力（Coupling）

9. 分解（Decomposition）

10. 企業解決方案或企業資源規劃（enterprise solutions software or enterprise resource planning（ERP）systems）

11. 環境（Environment）

12. 資訊系統分析與設計（Information systems analysis and design）

13. 介面（Interfaces）

14. 相關元件（Interrelated components）

15. 聯合應用設計（Joint Application Design, JAD）

16. 模組性（Modularity）

17. 委外（Outsourcing）

18. 參與設計（Participatory Design, PD）

19. 雛型法（Prototyping）

20. 目的（Purpose）

21. 快速應用程式發展（rapid application, RAD）

22. 儲存庫（Repository）

23. 系統（System）

24. 系統分析（Systems analysis）

25. 系統分析者（Systems analyst）

26. 系統設計（Systems design）

27. 系統發展生命週期（System development life cycle, SDLC）

28. 系統發展方法（Systems development methodology）

29. 系統實作與操作（Systems implementation and operation）

30. 系統規劃和選擇（System planning and selection）

重要辭彙檢測

從上述的關鍵名詞中選出最適合定義的選項。

(　　)1. SDLC 的第一個階段，分析與安排組織所需的資訊系統，而且確認可能的資訊系統計畫，並對計畫是否繼續的議題進行討論。

(　　)2. 藉由電腦資訊系統所發展和維護的複雜組織處理程序。

( )3. 一個源自於北歐的系統發展方法，核心的焦點在於使用者與他們的工作生活的改善。

( )4. 設計用來支援組織功能或程序的電腦軟體。

( )5. 承擔資訊系統的分析與設計大部份責任的組織角色。

( )6. 一個結構化的程序，使用者、管理者和分析師會因為一連串密集的會議而聚在一起數天，以確定或審查系統的需求。

( )7. 一個系統發展的反複程序，透過分析師和使用者之間密切的交流而不斷地修正，將需求被轉換成一個運作系統。

( )8. 一個元件間有互相關係的集合具有可識別的界限，為了某些目的而一起工作。

( )9. 一個不能縮減的部分，或是組成一個系統的部份聚集，也被稱為子系統。

( )10. 系統的一部份對其他系統中的一個或多個部分的依賴性。

( )11. 標示系統的內部與外部界線，而它將系統與環境區隔開來。

( )12. 一個系統的正體目標或功能。

( )13. SDLC 中一個階段，在系統分析中被選來發展的系統，先以獨立於任何的電腦平臺方式描述，然後再將可被完成的所有的程式和系統結構轉換成詳盡的技術規格。

( )14. SDLC 中的一個階段，研究目前的系統，並提出可供選擇的替換系統。

( )15. 在系統外部會與系統互相作用的所有事物。

( )16. 在的觀中系統與其環境的接觸點，或子系統彼此間的接觸點。

( )17. 系統成就上的限制。

( )18. SDLC 的最終階段，資訊系統將被編碼、測試和安裝在組織中，而且資訊系統也是有組織的被修補和改善。

( )19. 組織在進行分析、設計、實作以及維護資訊系統的必需步驟時，所遵循的一套標準程序。

( )20. 用來發展、維護和替換資訊系統的傳統方式。

( )21. 將系統的敘述分解成越來越小單元的反覆程序，這個程序會製作出一組圖表，而其圖表中的一個程序會在另一個圖表上作更加詳細的說明。

( )22. 將一個系統分成大小差不多的區塊或模組。

( )23. 子系統彼此相依的狀況。

( )24. 一個系統或子系統完成單一功能的程度。

( )25. 將公司的一些或全部的資訊系統應用和運作委託外面的公司負責。

( )26. 可針對部分系統發展過程提供自動化支援的軟體工具。

( )27. 一個集中的資料庫，它包含所以圖形、表格和報表的定義、資料結構、資料定義、流程和邏輯，以及其它組織和系統元件的定義，它提供一些機制和架構讓資料和工具間，以及資料和資料間可以整合。

( )28. 一個整合傳統企業功能至一系列模組的系統，以便讓一個交易作業發生在一個資訊系統而不是好幾個分散的系統。

( )29. 系統開發方法論在於減少時間需求來設計和實作資訊系統。

( )30. 現今的系統開發方法重點在於適性方法、人取代了角色和一個自我適應的開發程序。

複習問題

1. 什麼是資訊系統分析和設計呢？
2. 什麼是系統思考？它對電腦資訊系統上的考量有什麼幫助呢？
3. 什麼是分解？耦合？內凝？
4. 組織系統的規模是什麼？
5. 描述五種軟體資源。
6. 列出這一章資訊系統所描述的不同種類。它們彼此之間有何不同呢？
7. 列出並說明資訊系統發展生命週期的不同階段？
8. 什麼是雛型法呢？
9. 什麼是 CASE 工具？什麼是 CASE 的儲存庫和如何利用它？
10. 什麼是 JAD 呢？什麼是參與設計呢？
11. 什麼是 RAD？如何比較它與典型的 SDLC？
12. 什麼是敏捷方法論？

問題與練習

1. 當建立一個系統的時候，為什麼使用系統分析和設計的方法是重要的？為何任何建立系統的方法看起來都不像是“快速和容易的”呢？使用“工程”的方法有什麼樣的價值？

2. 以一個系統的觀點敘述你的大學或學院。什麼是系統的輸入呢？輸出呢？界限呢？元件呢？他們的相互關係呢？限制呢？目的呢？介面呢？環境呢？畫出這個系統的圖。

3. 一輛汽車就是一個系統與數個子系統，包括剎車子系統、電氣子系統、引擎、燃料子系統、氣候控制子系統與乘客子系統。繪出一部汽車的表示圖，如同一個系統並標註所有包的系統特性。

4. 你的個人電腦是一個系統。像你在問題和練習 3 所作的汽車一樣，將一部個人電腦繪製並標示成一個系統。

5. 選擇一個你平常從事的商業交易，例如使用一個 ATM 機器，在超級市場買食品雜貨，或為了一個大學的籃球賽買票。請為這個交易定義資料、繪出資料流程圖以及和描述處理的邏輯。

6. 從軟體雜誌中找到現今前 500 家軟體公司，並與表 1-1 比較有哪些公司的排名有改變？嘗試說明為何你列出來的名單不同於表 1-1，電腦產業有什麼改變而影響這名單？

7. 在北歐，聯合應用設計（JAD）的方法和參與設計（PD）的方法有什麼不同？你必須在圖書館作一些挖掘的工作，適當的回答這個問題。使用類似這個方法去建立資訊系統的利益是什麼呢？又有業什麼障礙呢？

8. 你如何組織一個學生的專案團隊與一個小型的企業委託人工作呢？假如你為一個專業的諮訊組織工作，你將如何組織一個專案團隊呢？這兩種組織方法的差異為何？為什麼呢？

9. 如何使用雛型法作為 SDLC 的一部份呢？

10. 敘述系統分析者的角色在 SDLC 與雛型法之間的不同。

11. 比較圖 1-10 與 1-11。你看到什麼樣相似和相異的部分呢？

討論問題

1. 假如某人在一個聚會中問你，系統分析者是什麼和為什麼任何人都想要成為他們中的一員，你將如何回答呢？從這一章中找出支持你答案的證據。

2. 說明以電腦資訊系統是如何設計去處理薪資這樣一個特定系統的例子。確定在你的解釋中說明任何系統中所有的九個元件。

3. 網際網路，特別是全球資訊網，為何適合用來說明系統分析和系統發展？

4. 你認為系統分析和設計在未來的十年狀況如何？如果你早一點看見這一章，系統發展的改變在最近已經相當的引人注意了。一位電腦的程式設計人員意外的從 1950 年代被傳送到 2000 年，這將會有一些確認電腦環境的麻煩，因為他的成長晚了近 50 年。什麼樣引人注意的改變將發生於未來的 10 年呢？

案例問題

1. 松谷傢俱

Alex Schuster 是以一位愛好者的身分創立了松谷傢俱。最初，Alex 在他的車房為朋友們與家人建造訂製的傢俱。關於他的工藝品資的消息傳開來了，他就開始接受訂單。這位業餘愛好者從此逐步轉變成一家中型的企業，並擁有超過 50 位員工。在這些年中，增加的需求迫使 Alex 重新定位數次，增加他的推銷人員，擴展他的生產線，以及重做松谷傢俱的資訊系統。當這家公司開始成長，Alex 將這家公司組織成為數個功能領域—製造、銷售、訂單，會計以及採購。起初，手工式的資訊系統被使用，然而隨著交易快速的擴張，迷你電腦被安裝用於自動化的應用。在最初，程序導向的方式經常被利用。每一個分離的應用程式有它們自己的資料檔案。這些應用程式模仿手工操作系統並使之自動化。一個改善它的資訊系統的成果，PVF 最近修復了它的資訊系統，並導致公司廣泛的資料庫和應用與該資料庫一同工作。松谷傢俱的電腦基礎應用主要是在會計和財務的方面。所有的應用已經被內部建立好了，並當需要的時候，新的資訊系統職員會被僱用去支援松谷傢俱擴展中的資訊系統。

a. 松谷傢俱是如何被開發它的資訊系統呢？為什麼你認為這家公司會做這樣的選擇呢？有什麼樣其他的選擇是可被利用的呢？

b. 一個可利用的選擇是公司廣闊的系統。公司廣泛系統的特色是什麼，是如同 SAP，或提供的功能嗎？企業廣泛系統的優點又是什麼呢？

c. 松谷傢俱將在下個用僱用二位系統分析者。你的任務是為了這些立場發展工作的宣傳。並為了系統分析者在數個網站或（和）報紙刊登工作的宣傳。以及需要什麼樣的技能？

d. 松谷傢俱現在使用什麼樣的資訊系統型態？為每一個提供例子。

2. 湖西爾漢堡

如同 1970 年代的大學生一樣，Bob 和 Thelma Mellankamp 經常夢想開始屬於他們自己的企業。當他們去上一門經濟學課程的途中，Bob 和 Thelma 駕駛車輛經由 Myrtle 的家庭餐廳並且注意到窗戶上面掛著“待售”的牌子。Bob 和 Thelma 迅速地準備購買這家企業，於是湖西爾漢堡餐廳就此誕生。這家餐廳是有限制的大小，由廚房、餐廳、櫃檯、儲存區與辦公室組成。現在，所有的書面作業都是由手動完成。Bob 和 Thelma 已經討論過購買電腦系統的利益；然而，Bob 想要調查一些選項以及僱用一個顧問來幫助他們。

易腐爛的食物項目，諸如：牛肉餡餅、小圓麵包以及蔬菜是每日被運送到這家餐廳。其他的項目，例如餐巾、吸管和杯子是當需求的時候才以訂單的方式運送。Bob Mellankamp 從餐廳的後門接收運送的貨物然後更新庫存的日誌表格。這庫存日誌表格幫助 Bob 追蹤庫存項目。庫存日誌表格被更新，是當運送的貨物被接收以及每夜清點每日的銷售總額時。顧客放置他們的菜單在櫃檯上，並且當他們所點的餐被準備好時被叫喚。

訂購的食物被寫在菜單上，並總計登記的金額，然後，將它傳遞給廚房，在那裡訂購的菜是已經準備好的。收銀機是沒有能力去取得銷售點的資訊。一旦所點的餐點被準備好和運送，菜單便被放置於菜單盒中。Bob 在晚上回顧這些菜單並做庫存的調整。

在最近的幾個月中，Bob 已經注意到湖西爾漢堡現在資訊系統的一些問題，特別是庫存的控制方面、顧客的點餐與管理報告系統。因為庫存控制和顧客點餐系統是紙上作業，錯誤的發生相當頻繁，通常會影響從供應商和顧客取得餐點的傳送菜單接收。Bob 經常想要有電子式的存取預測資訊、庫存使用率和基本的銷售資訊。對於紙上作業的系統來說，這樣的存取是不可能的。

a. 應用 SDLC 的方法到湖西爾漢堡

b. 使用湖西爾漢堡的腳本，確定每一個系統特性的例子。

c. 分解湖西爾漢堡進入它的主要子系統。

d. 簡短地總結第 1 章所討論的系統發展的方法。那一個方法你覺得應該被使用在湖西爾漢堡？

3. 自然健康食品百貨

自然健康食品百貨是連鎖有益健康的百貨服務於 Oklahoma，Arkansas 和 Texas。Garrett Davis 在 1975 年開創他的第一家自然健康食品百貨，到現在為止已經在三個州開了 15 間百貨。起初，他只銷草本的補給品，美食家咖啡和茶以及家用產品。在 1990 年，他擴展他的產品線到包含個人關心的食品，寵物的食品與南北貨的項目。

在過去幾個月中，Davis 先生的許多顧客已經請求去購買包裝好的熟食的能力，例如雞、火雞、魚和素食，這些已經包裝好的熟食會自動地運送到他們的家一週一次，兩週一次或一個月一次。Davis 先生感覺那是一個可實行的選，因為自然健康食品百貨有一個為了它存在的產品系列的自動運送系統。

在現在這個系統，顧客能夠署名給自然健康食品百貨運送服務（NBDS），以及個人關心的食品、寵物食品、美食產品和食品雜貨店在運送一週一次，兩週一次或以一個月為基礎。全部的訂購流程大約花費五分鐘。售貨員獲得顧客的姓名，郵寄地址，信用卡號通，希望運送的項目與數量，運送的頻率和電話號碼。在顧客的訂購已經被處理之後，通常會在一週內開始運送。就像顧客的訂單被處理，庫存會自動的更新。這個 NBDS 系統是一個主從式系統。每一個百貨店會被裝置一個客戶端電腦，並存取離開中心的資料庫資料去覆蓋中央伺服器。伺服器端追蹤庫存資料，顧客的活動，運送的時程表，與個別商店的銷售額。每星期 NBDS 會產生銷售總額報表，最低庫存的報表以及每個商店的運送時程表。這個資訊包含這些中每一個個別的報告然後合併到主要的銷售總結，最低庫存的報表和預測報表。這些報告所包含的資訊幫助重新進貨，產品的運送以及預測決定。Davis 先生有一個 Excel 的工作表，他使用來合併每一個商店的銷售資訊。然後他使用這個工作表為每一個商店作預測的決定。

a. 確定自然健康食品百貨所使用的不同資訊系統的型態？並為每一個系統提供一個例子？專家系統現在能被使用嗎？如果不行，在這一章所繪製的圖案？自然健康食品百貨如何從使用中的系統中獲利呢？

b. 圖 1-4 確定系統的七個特性。使用自然健康食品百貨的腳本，為每一個系統特性提供一個例子。

c. 自然健康食品百貨所擁有的電腦環境的工作型態是什麼？