

# 第一章 資訊系統開發概論

---



# 內容大綱

## 學習目標

- 1.1 導論
- 1.2 資訊系統開發的原則
- 1.3 資訊系統開發的構面
- 1.4 資訊系統開發的特性與挑戰
- 1.5 資訊系統開發環境
- 1.6 結論



# 學習目標

詳讀本章，你至少能瞭解：

- ❑ 系統開發、資訊系統開發和系統分析與設計的關係。
- ❑ 影響資訊系統開發的重要構面。
- ❑ 資訊系統開發環境。
- ❑ 資訊系統開發有哪些重要人員參與，各扮演何種角色。
- ❑ 資訊系統之建置有哪些策略可思考。
- ❑ 有哪些資訊系統開發模式。
- ❑ 系統分析與設計階段有哪些技術可用。



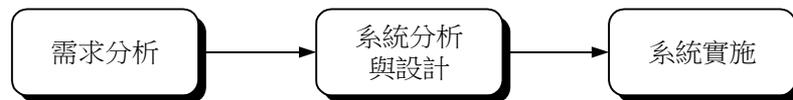
## 1.1 導論

- ✘ 隨著資訊科技的創新、資訊科技應用的普及和經營環境之競爭，有效地掌控與應用資訊已成為企業成敗的關鍵因素。
- ✘ 為能有效地管理與應用資訊，以支援組織的經營管理和決策需求，各種資訊系統乃應運而生：
  - + 交易處理系統
  - + 管理資訊系統
  - + 企業資源規劃系統
  - + 決策支援系統
  - + 專家系統
  - + 高階主管資訊系統等

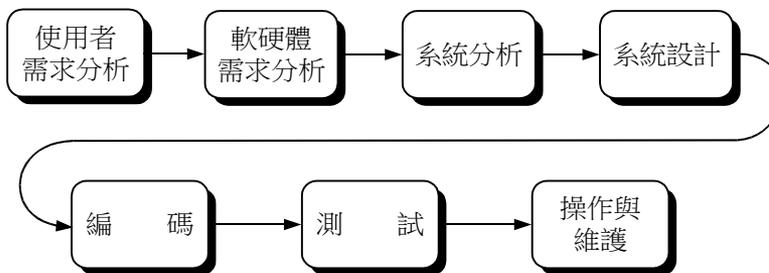


## 1.1 導論 (c.2)

- 雖然資訊系統各有其不同特性，系統開發的策略與技術也不盡相同，但系統開發過程仍可大略歸納出一些基本而共同步驟或階段
- 較單純之系統：分為需求分析、系統分析與設計、系統實施三階段



- 較複雜之系統：可劃分為七個階段或更多階段



- 無論如何劃分，「系統分析與設計」都是重要的開發階段之一



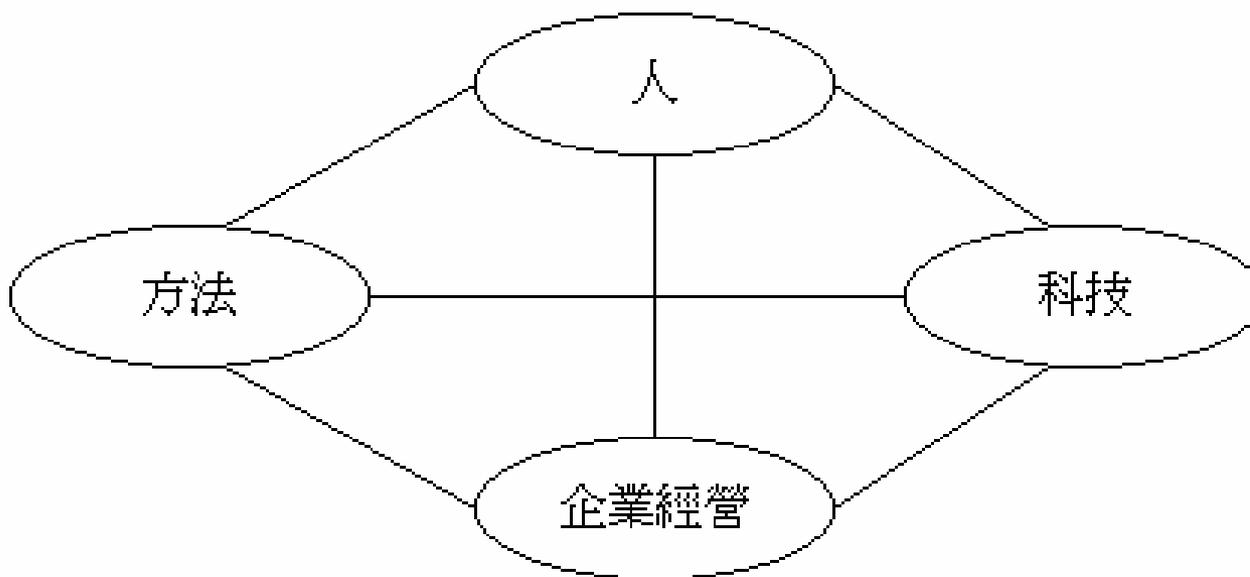
## 1.2 資訊系統開發的原則

- 資訊系統開發可定義為運用資訊科技及資訊系統開發方法，來建構實體的或邏輯的系統，以達到特定的目標或功能，協助人們解決資訊處理的需求。
- 成功的資訊系統開發必須把握一些種要的原則：
  - 系統的目標應該明確定義
  - 系統的目標要實際而有用
  - 系統開發要充分運用科技
  - 系統開發要依循一定的方法
  - 系統開發必須獲得足夠的資源與支持
  - 系統開發必須符合限制條件
  - 系統開發必須考慮環境因素



## 1.3 資訊系統開發的構面

- 資訊系統開發要考慮人、方法、科技和企業經營等四個重要的構面，其關係有如鑽石結構，各構面須均衡發展：



## 1.4 資訊系統開發的特性與挑戰

- 資訊系統開發的特性如下：
  - 軟體是人造的邏輯系統。
  - 軟體開發是一種專業的工作。
  - 軟體開發是一種智力活動。



## 1.4 資訊系統開發的特性與挑戰 (c.2)

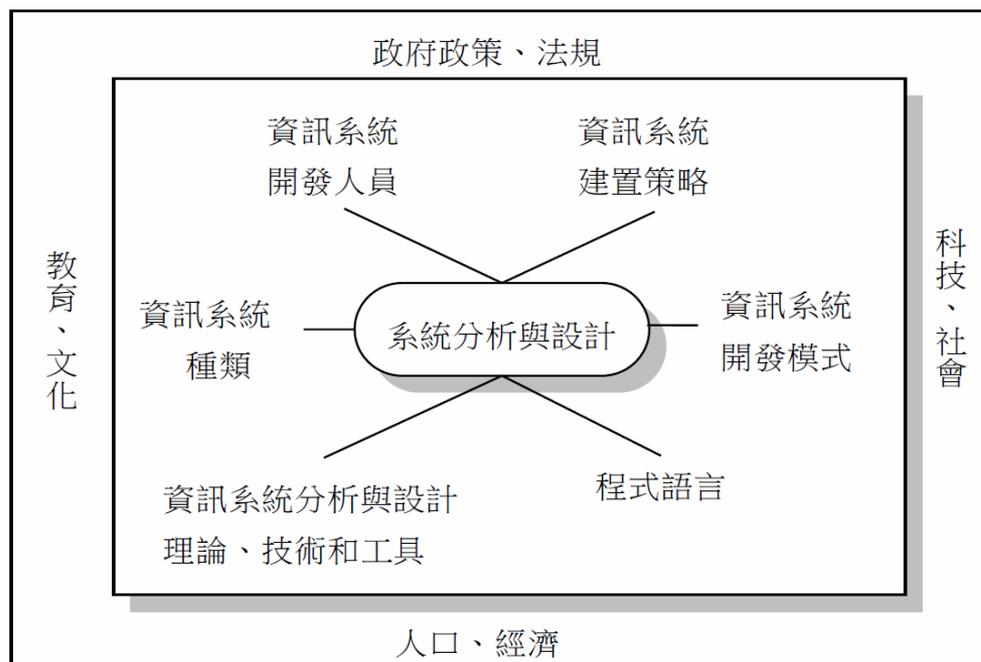
### □ 資訊系統開發面臨之挑戰：

- 資訊科技的快速進步，使得淘汰率驟增、系統的可用壽命變短、技術支援無法持久。
- 經營環境之快速變遷，使得需求亦經常變動。
- 資訊系統開發經常面臨時程延誤、成本超支、品質不良等問題。
- 人們重視看得見的硬體，而常忽略看不見的軟體。



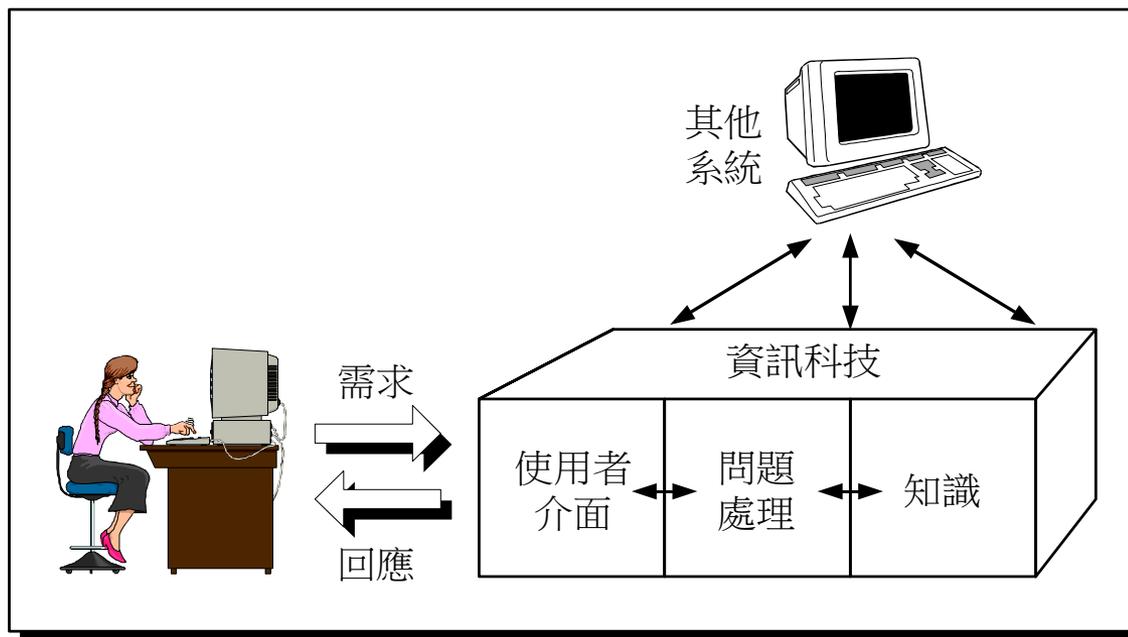
## 1.5 資訊系統開發環境

- 資訊系統開發環境所涉及的層面很廣，包括資訊系統的開發模式、種類、建置策略、開發人員、系統分析與設計技術等。
- 外在大環境的科技、社會、教育、文化、人口、經濟、政府政策與法規等因素，也直接或間接影響系統分析與設計。



## 1.5.1 資訊系統的種類

- 資訊系統是由使用者介面、問題處理與知識等元件，利用資訊科技有組織的結合，以管理（如擷取、傳輸、儲存、處理與展示等）知識，並為某些活動（如企業流程或問題解決等）提供知識或服務的系統。



## 1.5.1 資訊系統的種類 (c.2)

- 資訊系統已成為企業必備之工具，常見的資訊系統有：
  - 交易處理系統 (Transaction Processing System)
  - 管理資訊系統 (Management Information System)
  - 決策支援系統 (Decision Support System)
  - 高階主管資訊系統 (Executive Information System)
  - 專家系統 (Expert System)
  - 企業資源規劃 (Enterprise Resource Planning, ERP) 系統



# 交易處理系統

- 交易處理系統 (Transaction Processing System) 亦稱資料處理系統 (Data Processing System, DPS)
  - 主要目的是將大量的交易處理自動化。這類交易處理非常結構化，其程序及資料有既定的標準，且有高度重複性。
  - 此系統有兩種主要功能：
    - 交易紀錄之保存
    - 交易表單之產生
  - 銷售點管理 (Point of Sale, POS) 系統之前檯系統、加油站之加油作業與收銀系統、金融機構之櫃檯系統等皆屬交易處理系統。



# 管理資訊系統

## □ 管理資訊系統 (Management Information System, MIS)

- 主要目的為提供不同層級的管理者，有關組織營運狀況但摘述程度不同之報表，這些報表格式已事先訂定，亦即這些資料之處理與報表產生大多為結構化。
- 該系統主要有兩種功能：
  - 交易紀錄之保存
  - 摘述性報表之產生
- POS系統之後檯系統便屬於管理資訊系統。



# 決策支援系統

## □ 決策支援系統 (Decision Support System, DSS)

- 主要目的為支援決策者，提升其決策效率 (Efficiency) 與效能 (Effectiveness)。
- 決策支援系統主要支援半結構化或非結構化之決策活動，其特徵有：
  - 能以即興 (Ad Hoc)、自訂或標準化的方式分析資料與產生報表。
  - 能直接與決策者互動。
- POS 之後檯系統，除了固定式地資料查詢、處理、分析與報表產生外，若還能與使用者互動，並依其需求擷取、分析與展示資訊，則該系統可稱決策支援系統。



## 高階主管資訊系統

- 高階主管資訊系統 (**Executive Information System, EIS**)，亦稱高階主管支援系統 (**Executive Support System, ESS**)
  - 針對高階主管之資訊需求而設計
  - 希望高階主管能即時由系統獲得其所需之關鍵資訊，不需透過第三中介者整理或轉達。
  - **EIS** 之特徵包括可過濾、摘述關鍵資訊，亦可將資訊以多種方式作圖形化的展示 (如儀表板、長條圖)，也得以由上而下之方式擷取資訊並進行分解。
  - 一般而言，**EIS** 之特徵與**DSS** 相同，但比**DSS** 有更多限制，故高階主管資訊系統可視為**DSS** 的一種特例。



# 專家系統

## □ 專家系統 (Expert System, ES)

- 初期發展目的是用以模仿人類專家解決特定問題之能力，並希望專家系統所提供之答案或建議可達人類專家水準。
- 專家系統包含三個主要元件：
  1. 使用者介面：為專家系統與使用者交談之媒介。
  2. 推理引擎：為專家系統依使用者要求，由知識庫推論出結果或建議之機制。
  3. 知識庫：為系統儲存專家知識之處。
- 現今，專家系統不再強調取代專家，而是支援專家，因此專家系統也漸成為另一種形式之決策支援系統。



# 企業資源規劃系統

## □ 企業資源規劃 (Enterprise Resource Planning, ERP) 系統

- 為一種將多項企業功能整合在一起的模組化與架構化套裝資訊系統，藉由及時整合與掌握企業分散於各地的資源，提供最佳流程典範 (Best Practice)，以降低企業營運成本並提升客戶服務滿意度。
- 狹義的觀點視ERP 系統為整合企業內部資源規劃之系統。
- 廣義的觀點視ERP 系統為整合企業內外部資源規劃之系統，例如企業上下游之供應鏈管理等。



# APP

- **APP** 為「應用」(Application)的縮寫，原係指電腦上專門解決使用者需求所開發、撰寫的應用程式或軟體。
- 隨著行動裝置的興起，**APP**亦可指軟體開發商在行動裝置上開發的各種擁有獨立操作系統之行動軟體應用程式 (Mobile Application)。
- **APP**的功能主要圍繞於現代人日常生活的需求，包含：社交軟體影音娛樂、商業工具、遊戲、交通資訊等，使用者可以依照個人需求在軟體應用程式商店中下載並安裝於行動裝置中。



# 資訊系統之種類及其特性

資訊系統種類	資訊系統特性
交易處理系統	<ul style="list-style-type: none"> <li>➤ 將大量的交易處理自動化</li> <li>➤ 處理程序與資訊需求非常結構化</li> </ul>
管理資訊系統	<ul style="list-style-type: none"> <li>➤ 提供不同層級之管理者摘述程度不同之報表</li> <li>➤ 資料的處理與報表的產生亦多為結構化</li> </ul>
決策支援系統	<ul style="list-style-type: none"> <li>➤ 支援決策者半結構化或非結構化之決策</li> <li>➤ 提升決策之效率與效能</li> </ul>
高階主管資訊系統	<ul style="list-style-type: none"> <li>➤ 支援高階主管即時瞭解所需之關鍵資訊</li> <li>➤ 可過濾、摘述關鍵資訊</li> <li>➤ 可將資訊以多種方式作圖形化的展示</li> </ul>
專家系統	<ul style="list-style-type: none"> <li>➤ 針對特定應用範圍或領域，集合不同專家知識而成的系統</li> <li>➤ 希望專家系統提供之答案或建議可達人類專家之水準</li> </ul>
企業資源規劃系統	<ul style="list-style-type: none"> <li>➤ 將多項企業功能整合在一起</li> <li>➤ 可即時整合與規劃企業分散於各地之資源</li> <li>➤ 提供最佳的流程典範</li> </ul>
APP	<ul style="list-style-type: none"> <li>➤ 針對行動裝置所開發的各種微型應用程式</li> <li>➤ 擁有獨立的操作系統</li> <li>➤ 使用者透過軟體應用程式商店直接下載並安裝於行動裝置</li> </ul>

## 1.5.2 資訊系統開發相關人員之角色

- 資訊系統開發之相關人員包括
  - 終端使用者 (End User)
  - 終端使用者之高層主管
  - 程式設計師 (Programmer)
  - 系統分析師 (System Analyst)
  - 資訊系統經理 (IS Manager)
  - 資料庫管理者 (Database Administrator)
  - 其他技師 (Technician) 等。



## 1.5.2 資訊系統開發相關人員之角色 (c.2)

### □ 系統分析師

- 擷取使用者需求，並能清楚且完整地瞭解及表達需求。
- 進一步將需求轉換成資訊技術、企業處理與知識等元件，並將之有組織地結合起來。

### □ 程式設計師

- 主要是依分析與設計之藍圖設計出程式、建立資料庫、測試與安裝系統等。
- 系統分析與設計的過程中，必須和各種人員保持良好的溝通、互動與互信，以確保系統開發之成功。



## 1.5.2 資訊系統開發相關人員之角色(c.3)

### □ 終端使用者

- 是問題領域之專家，但可能並非資訊科技方面之專家，其於系統分析與設計過程中，主要扮演提供使用者需求與企業知識的角色。
- 因此，終端使用者應能主動、積極地參與系統開發，並盡可能完整與清楚地表達需求和知識，而非被動地等待系統分析師之詢問。
- 系統分析師要與終端使用者一起工作，並將其需求和企業上的知識轉換成可支援終端使用者工作的資訊系統。



## 1.5.2 資訊系統開發相關人員之角色(c.4)

### □ 企業經理

- 是終端使用者之高層主管，這些人沒有很多的時間參與系統的開發，但他們在系統開發的過程中扮演了非常重要的角色，例如承諾對系統開發所需之財力、人力、時間等資源的投入及提出組織之政策與限制等。
- 由於他們的決策權與對企業經營的專業知識，部門的領導者與執行長也能對資訊系統專案發展設定一般性的需求與限制。
- 系統分析師必須要這些人保持良好的溝通，瞭解他們的期望和優先順序，並從他們口中瞭解組織內的限制和資源。



## 1.5.3 資訊系統建置策略

- ✘ 資訊系統建置策略是指新資訊系統之建立，與舊資訊系統之修改、擴充或更新等所採取的方式。
- ✘ 資訊系統之建置策略可分成三種：
  1. 由公司內部自行完成
  2. 由公司外部取得
  3. 其他方式
- ✘ 若採取由公司內部自行完成之策略
  - ✚ 即使用者自建 (**End User Computing**)
    - ✘ 由公司資訊部門自行開發
    - ✘ 由相關部門人員組成任務編組開發



## 1.5.3 資訊系統建置策略(c.2)

- 若取自於公司外部，則可進行：
  - 委外開發 (**Outsourcing**)
  - 購買現成之套裝軟體 (**Application Package**)
  - 引進同業之系統
  - 採取租賃方式，由應用系統服務提供者 (**Application Service Provider, ASP**) 提供軟體或硬體服務
  
- 其他方式
  - 上述各種策略之綜合
  - 由同業聯合共同尋找資訊公司開發



## 1.5.4 資訊系統開發模式

- 系統開發模式主要考量系統開發過程應分成哪些階段、每個階段應如何進行及包含哪些工作內容等議題。
- 常用的資訊系統開發方法有八種：
  - 瀑布模式 (Waterfall Model)
  - 雛型模式 (Prototyping Model)
  - 漸增模式 (Incremental Model)
  - 螺旋模式 (Spiral Model)
  - 同步模式 (Concurrent Model)
  - RUP模式 (Rational Unified Process Model)
  - 敏捷軟體開發 (Agile Software Development)
  - MDA軟體發展生命週期 (Model Driven Architecture)



## 1.5.5 程式語言

- 程式語言從1950年代的機器碼 (**Raw Machine Code**) 逐漸演進為組合語言 (**Assembly Languages**)、程序性程式語言 (**Procedural Programming Languages**) 到現在的物件導向程式語言 (**Object-Oriented Programming Languages**)

年代	當時的程式語言
1950	機器碼
1950~1965	組合語言
1965~1985	程序性程式語言
1985~現在	物件導向程式語言
現在~ .....	下一波為何？



## 1.5.5 程式語言(C.2)

- 程式語言從1950年代的機器碼 (**Raw Machine Code**) 逐漸演進為組合語言 (**Assembly Languages**)、程序性程式語言 (**Procedural Programming Languages**) 到現在的物件導向程式語言 (**Object-Oriented Programming Languages**)

年代	當時的程式語言
1950	機器碼
1950~1965	組合語言
1965~1985	程序性程式語言
1985~現在	物件導向程式語言
現在~ .....	下一波為何？



## 1.5.6 系統分析與設計理論、技術和工具

- 系統分析與設計是一系列有組織之處理程序，目的是將使用者或企業需求轉換成有組織的資訊科技、企業流程與知識等元件。
- 常用的系統分析與設計技術可分為兩種：結構化技術與物件導向技術。
  - 結構化技術主要用於系統開發過程之分析與設計階段，幫助系統分析師進行資訊系統之描述與驗證。
  - 過程中將企業流程與資料分開處理。
  - 以結構化塑模工具幫助系統分析師進行資訊系統之描述與驗證。



## 1.5.6 系統分析與設計理論、技術和工具 (C.2)

- 應用結構化技術時，常用之流程與資料塑模(**Modeling**)工具有：
1. 事件(Event)
  2. 環境圖(Context Diagram)
  3. 資料流程圖(Data Flow Diagram, DFD)
  4. 資料字典(Data Dictionary, DD)
  5. 處理規格描述(Process Specification)
  6. 實體關係圖(Entity-Relationship Diagram, ERD)



## 1.5.6 系統分析與設計理論、技術和工具 (C.3)

- 物件導向技術是將企業流程與資料封裝成物件，而不像結構化技術是將企業流程與資料分開處理。
- 常用的塑模工具為**統一塑模語言**。
- 物件導向之另一特徵是繼承的概念，也就是當物件類別之層級結構被建立時，低層物件可繼承高層物件之性質，以減少因需求改變而造成相關程式之修改。



## 1.5.6 系統分析與設計理論、技術和工具 (C.4)

□ 統一塑模語言 (Unified Modeling Language, UML) 共使用二十二種模式圖進行塑模：

1. 使用個案圖
2. 資訊流圖
3. 活動圖
4. 狀態機圖
5. 行為狀態機圖
6. 協定狀態機圖
7. 循序圖
8. 溝通圖
9. 時序圖
10. 互動概觀圖
11. 類別圖
12. 物件圖
13. 套件圖
14. 模型圖
15. 複合結構圖
16. 內部結構圖
17. 合作使用圖
18. 元件圖
19. 部署圖
20. 表現圖
21. 網路架構圖
22. 輪廓圖



## 1.5.7 外在環境

- 外在環境指資訊系統開發所面對的組織、社會、文化、教育、科技、法規、政策等因素，這些因素直接或間接影響到資訊系統開發的成效。
  - 資訊密集行業如金融業；強調決策速度之環境如科技產業；勇於嘗試新科技的組織文化、人員教育程度較高的組織、重視智慧財產權的社會價值觀等特性，都有利資訊系統的發展。
  - 科技的成熟度、科技取得的方便性、科技人員的普及性、法規的要求或同業的協定，如金融機構之跨行連線等，也是有利的外在環境因素。
  - 此外，社會媒體的宣導、政府政策的推動、智慧財產權的保護、資訊教育的推廣等，均有利資訊系統開發。



## 1.6 結論

- 資訊系統開發為有系統、有組織的活動，過程中需考量系統開發之環境背景，如瞭解資訊系統種類、開發人員、建置策略、開發模式、程式語言及分析與設計技術等，並做最適當的選擇。
- 選擇時，也應考量組織之限制、政策、人員之資訊科技能力、資金、規模、對系統需求之迫切性、組織對資訊倚賴之程度與重要性等。
- 資訊人員應對上述問題有充分地瞭解，才能有效地掌握系統開發之進行。

