



# 第一章

## 系統分析設計概觀



# 大綱

1-1 認識系統分析與其重要性

1-2 軟體發展程序

1-3 系統分析師與系統分析設計之工作

## 1-1 認識系統分析與其重要性

# 軟體系統的普遍性

- 在當今環境中，軟體已經無所不在



©Tomozo/Flickr (CC BY-SA 2.0)



©smoothgroover22/Flickr (CC BY-SA 2.0)



©MaurizioPesce/Flickr (CC BY-SA 2.0)

**圖 1.1** 改變人類生活的應用：機器人、自駕車、虛擬實境

- 手機及汽車之嵌入式軟體
- 企業內部的各種管理資訊系統
- 高速公路的收費系統、悠遊卡系統、通關系統

# 軟體系統可能遇到的問題

- 軟體系統出現最多錯誤的環節在系統分析設計階段
- 主要問題：
  - 使用者對於需求無法明確說明
  - 環境變動太快導致經常更動需求
  - 開發人員專案管理不良、不瞭解正確發展方法與成本估算過低的問題，使得發展出來的產品品質不良而造成嚴重錯誤

# 系統分析與設計工作

- 有效解決軟體可能引發的問題
  - 重視軟體發展時的程序控管與品質要求
  - 做到更為完善的系統分析與設計工作
- 所謂系統分析，是指分析師去瞭解和明確敘述出資訊系統應該完成之工作的一組活動
- 所謂系統設計，是指分析師定義和描述系統解決需求細節的一組活動

# 軟體系統可能遇到的問題

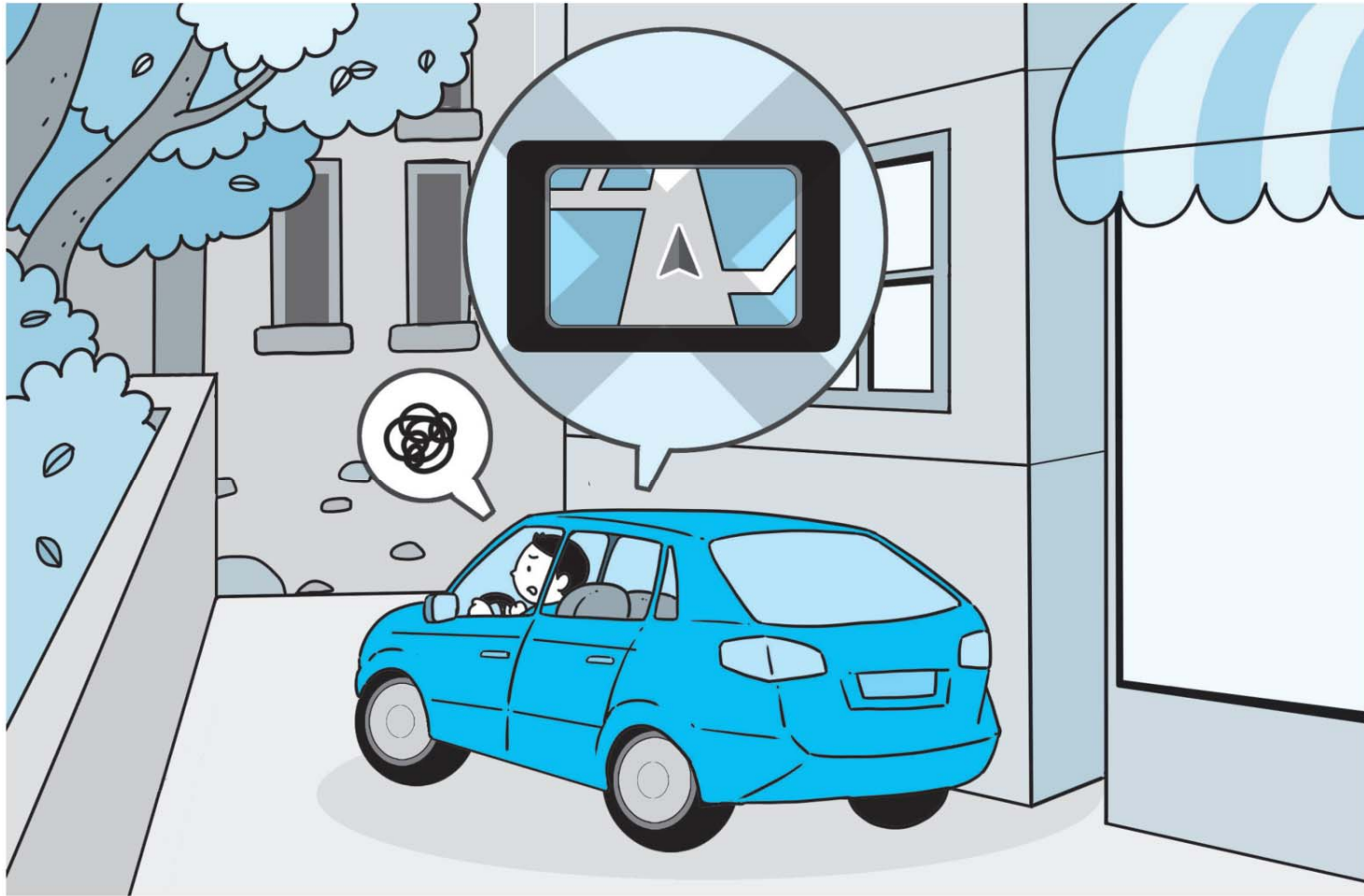


圖 1.2 系統發展的問題造成系統運作錯誤



# 系統分析設計方法

- 目前大家普遍使用的系統分析設計**方法論**為
  - **結構化方法** ( structured methodology )
  - **物件導向方法** ( object-oriented methodology )



# 結構化方法

- 將結構化分析、結構化設計與結構化程式設計方法，整合成為一套完整之系統發展方法
- 強調系統發展要設計出好的結構，除了可以簡化設計外，未來維護工作也會變得容易

# 結構化方法工作重點與產出

**表 1.1** 結構化方法之工作重點與產出

階段	工作重點	工具	產出
分析	<ul style="list-style-type: none"> <li>▪ 收集與確認使用者需求</li> <li>▪ 撰寫系統規格</li> <li>▪ 邏輯資料庫設計</li> </ul>	<ul style="list-style-type: none"> <li>▪ 訪談、問卷、開會、觀察、研究等</li> <li>▪ 資料流程圖、資料字典、迷你規格書</li> <li>▪ 實體關係模型</li> </ul>	<ul style="list-style-type: none"> <li>▪ 系統需求之原始資料</li> <li>▪ DFD圖、資料字典和迷你規格書</li> <li>▪ 資料表關聯圖</li> </ul>
設計	<ul style="list-style-type: none"> <li>▪ 模組化設計</li> <li>▪ 模組規格撰寫</li> <li>▪ 實體資料庫設計</li> <li>▪ 介面設計</li> </ul>	<ul style="list-style-type: none"> <li>▪ 結構圖</li> <li>▪ 結構化英文或虛擬碼</li> <li>▪ 資料定義語言、資料描述語言</li> <li>▪ 介面設計工具和報表產生器</li> </ul>	<ul style="list-style-type: none"> <li>▪ 結構圖</li> <li>▪ 模組規格書</li> <li>▪ 實體資料庫</li> <li>▪ 介面規格</li> </ul>

# 物件導向方法

- 1994 年，Booch 和 Rumbaugh 兩人結合其發展方法，發展出物件導向系統所需要的標準程序，稱為**統一程序**（ unified process, UP ）
- 1995 年 Jacobson 加入了他們的工作，並在 1997 年發展出標準的物件模式化語言，稱為 **UML**（ Unified Modeling Language ）
- 1997 年 OMG（ Object Management Group ）採用 UML 當作物件導向方法標準的模式化工具
- 至此，物件導向方法論的發展已趨於成熟

# 物件導向方法工作重點與產出

- 將系統看成是不同的個體（object）組成，或者將這些個體稱為物件
- 每個物件中都擁有其自己的資料和程序
  - 物件導向方法上，程序稱之為操作或方法
- 系統分析師並不需要太關心物件本身的程序如何進行，而是要關心系統應由哪些個體組成
- 而系統所要解決的問題是由個體間的互動與資料（或是訊息）的傳遞來完成
- 物件導向方法具有一些基本特性，如物件封裝、繼承與多型的觀念

# 系統與物件

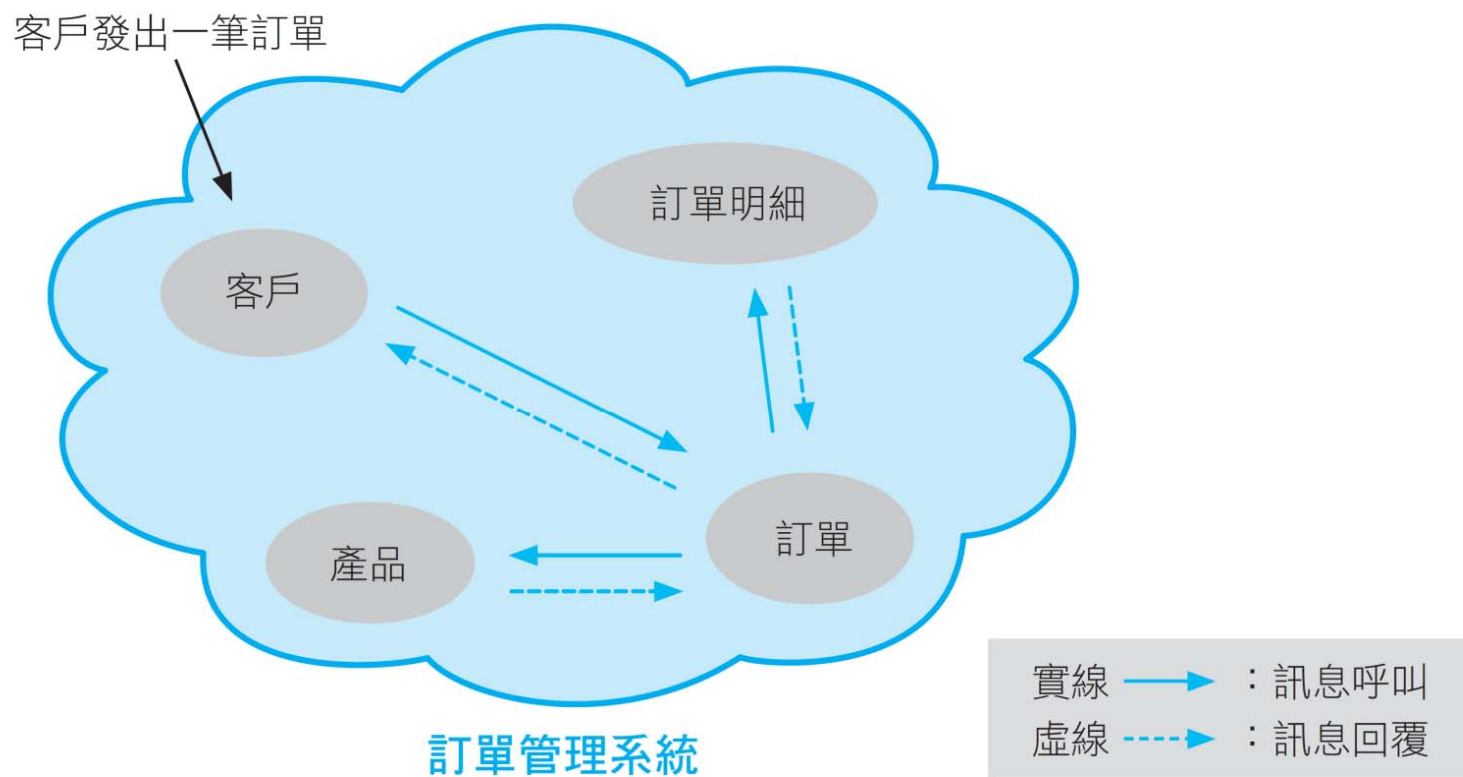


圖 1.3 系統與物件

## 1-2 軟體發展程序



# 規範式程序

- 將軟體發展分為幾個階段進行，每一個階段要完成的工作必須事先仔細定義好
- 上個階段執行完之後才執行下一個階段工作
- 每一階段完成後必須得到使用者的確認



# 軟體發展程序

- 軟體發展程序指的就是製造出軟體的方式
- 可以看成是在發展和評估軟體時所使用的一組活動、方法和實踐
  - 規範式程序 ( prescriptive process ) 或稱為計畫驅動程序 ( plan-driven process )
  - 敏捷式程序 ( agile process ) 或適應性程序 ( adaptive process )

# 敏捷式程序

- 強調在能夠快速回應使用者的需求改變與環境的變化，採用了**反覆與漸增式**的發展方法
- 強調專案的快速回應能力，必需要遵守一組原則，包括
  - **增量規劃** ( incremental planning )
  - **小版本發行** ( small releases )
  - **測試驅動** ( test-first driven )
  - **重構** ( refactoring )
  - **配對程式發展** ( pair programming ) 等

# 敏捷式程序之規範

表 1.2 敏捷式程序之規範

增量規劃	客戶的需求是以一個個增量模組來開發，開發後的增量模組將交付使用者使用，並取得使用者之回饋意見
小版本發行	最具重要性的功能優先開發，每次發行的功能不多，系統發行頻繁，並逐漸增加功能
測試驅動	先依據規格撰寫測試個案，程式撰寫過程不斷以這些測試個案測試，直到全部通過為止
重構	改善程式結構但不改變功能
配對程式發展	開發人員以兩人為一組，一人撰寫程式一人負責測試，約定一個時間後互換工作

# 瀑布式軟體發展程序

- 瀑布式軟體發展程序 ( waterfall software development process ) 屬於規範式程序
- 將軟體發展分成五個階段：需求、分析、設計、製作和維護
- 必須階段工作完成後才能繼續發展下一階段
- 某階段的工作完成後，必須凍結 ( freeze )，不允許再改變這一階段的內容

# 瀑布式軟體發展程序

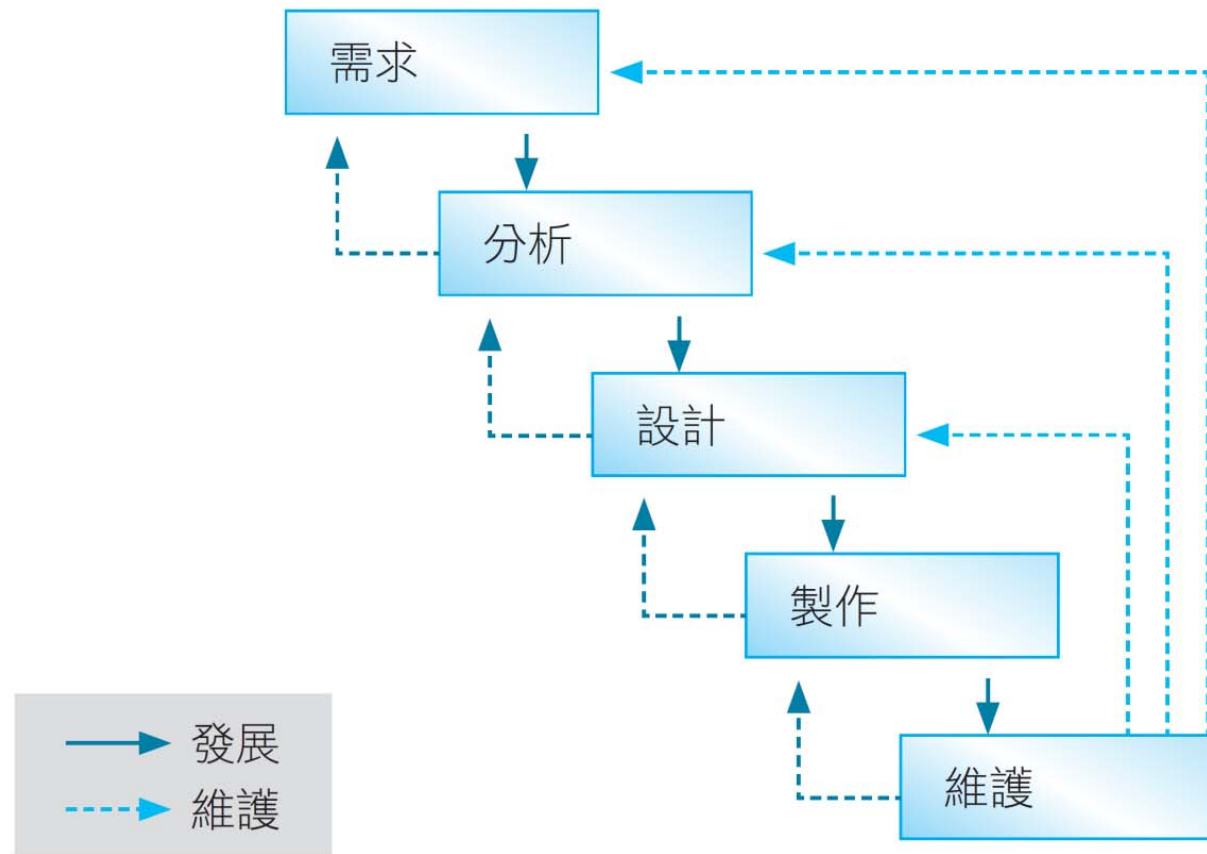


圖 1.4 瀑布式軟體發展程序

# 軟體發展程序五個階段

- 需求階段

- 使用者需求調查，瞭解使用者之目標與需求
- **功能性需求**（ functional requirement ）、**非功能性需求**（ non-functional requirement ）。

- 分析階段

- 使用系統規格書記錄使用者需求與限制，並得到使用者的認可後，才進入設計階段

# 軟體發展程序五個階段

- 設計階段

- 將規格書進一步分析，建立系統架構及細部模組，除了表現出模組間的關係外，還要對每個模組進行描述，稱為系統設計書



# 軟體發展程序五個階段

## ● 製作階段

### ➤ 分三部分，分別為程式撰寫、測試與上線

- ◆ 程式撰寫：將系統設計書轉為程式模組
- ◆ 測試工作：進行單元測試、整合測試、功能測試和系統測試
- ◆ 上線工作：將測試完成的系統擺放至實際環境中運作

## ● 維護階段

### ➤ 上線後所進行的工作

- ◆ 正確性維護是指修改系統錯誤所進行的工作
- ◆ 適應性維護是指因應環境變化所進行的修改工作
- ◆ 完美性維護則是使用者需要提升系統運作效能所做的調整工作
- ◆ 預防性維護用來預防未來環境改變時所進行的工作



# 統一程序

- 基本特性包括：
  - 反覆式發展
  - 反覆並漸增式的發展
  - 接受改變並調整與適應
  - 時間區間之概念



# 增量模組

- 為因應需求不確定與環境可能的變化，將系統規劃成不同的小部分稱為**增量模組**
- 每一個小的時間區間完成一個小增量模組，測試完成並交付使用者使用
- 使用後的意見在第二個時間區間進行調整，並進行第二個增量模組開發，直到系統完成為止

# 統一程序階段

- 共分四個階段，分別為
  - 起始階段 ( inception )
  - 詳細闡述階段 ( elaboration )
  - 建構階段 ( construction )
  - 轉換階段 ( transition )
- 每一階段都會經歷軟體發展生命週期的所有階段

# 統一程序工作重點與產出

**表 1.3** 統一程序之工作重點與產出

階段	工作重點	工具	產出
起始	<ul style="list-style-type: none"> <li>瞭解領域的問題、產品的範圍</li> <li>確認願景目標</li> <li>建立起始使用案例</li> <li>利害關係人同意投資在這個系統的開發</li> </ul>	<ul style="list-style-type: none"> <li>成本效益的分析</li> <li>風險評估</li> <li>使用案例</li> <li>領域模型</li> </ul>	<ul style="list-style-type: none"> <li>系統的願景</li> <li>成本效益分析</li> <li>初步使用案例</li> <li>補充的規格</li> <li>詞彙表</li> </ul>
詳細闡述	<ul style="list-style-type: none"> <li>強化系統架構</li> <li>風險控管</li> <li>強化起始之使用案例</li> <li>製作專案管理計畫</li> </ul>	<ul style="list-style-type: none"> <li>使用案例</li> <li>領域模型</li> <li>互動圖</li> <li>類別圖</li> <li>程式語言</li> </ul>	<ul style="list-style-type: none"> <li>企業領域模型</li> <li>使用案例</li> <li>設計模型（互動圖、類別圖等）</li> <li>資料模型（E-R model）</li> <li>程式製作</li> <li>測試計畫</li> <li>專案管理計畫</li> </ul>

# 統一程序工作重點與產出

表 1.3 統一程序之工作重點與產出

建構	<ul style="list-style-type: none"><li>發展出第一個可作業的軟體系統版本</li><li>單元測試、整合測試和整體系統的功能測試</li></ul>	<ul style="list-style-type: none"><li>使用案例</li><li>領域模型</li><li>互動圖</li><li>類別圖</li><li>程式語言</li><li>測試方法</li></ul>	<ul style="list-style-type: none"><li>設計模型</li><li>程式製作</li><li>資料模型</li><li>測試計畫</li><li>專案管理計畫</li></ul>
轉換	<ul style="list-style-type: none"><li>確認滿足客戶需求</li><li>更正錯誤</li><li>完成所有文件</li><li>發現風險</li><li>客戶回饋修正</li></ul>	<ul style="list-style-type: none"><li>使用案例</li><li>領域模型</li><li>互動圖</li><li>類別圖</li><li>程式語言</li><li>測試方法</li></ul>	<ul style="list-style-type: none"><li>完整的系統</li><li>所有的發展文件</li></ul>

# 統一程序階段與軟體生命週期關係

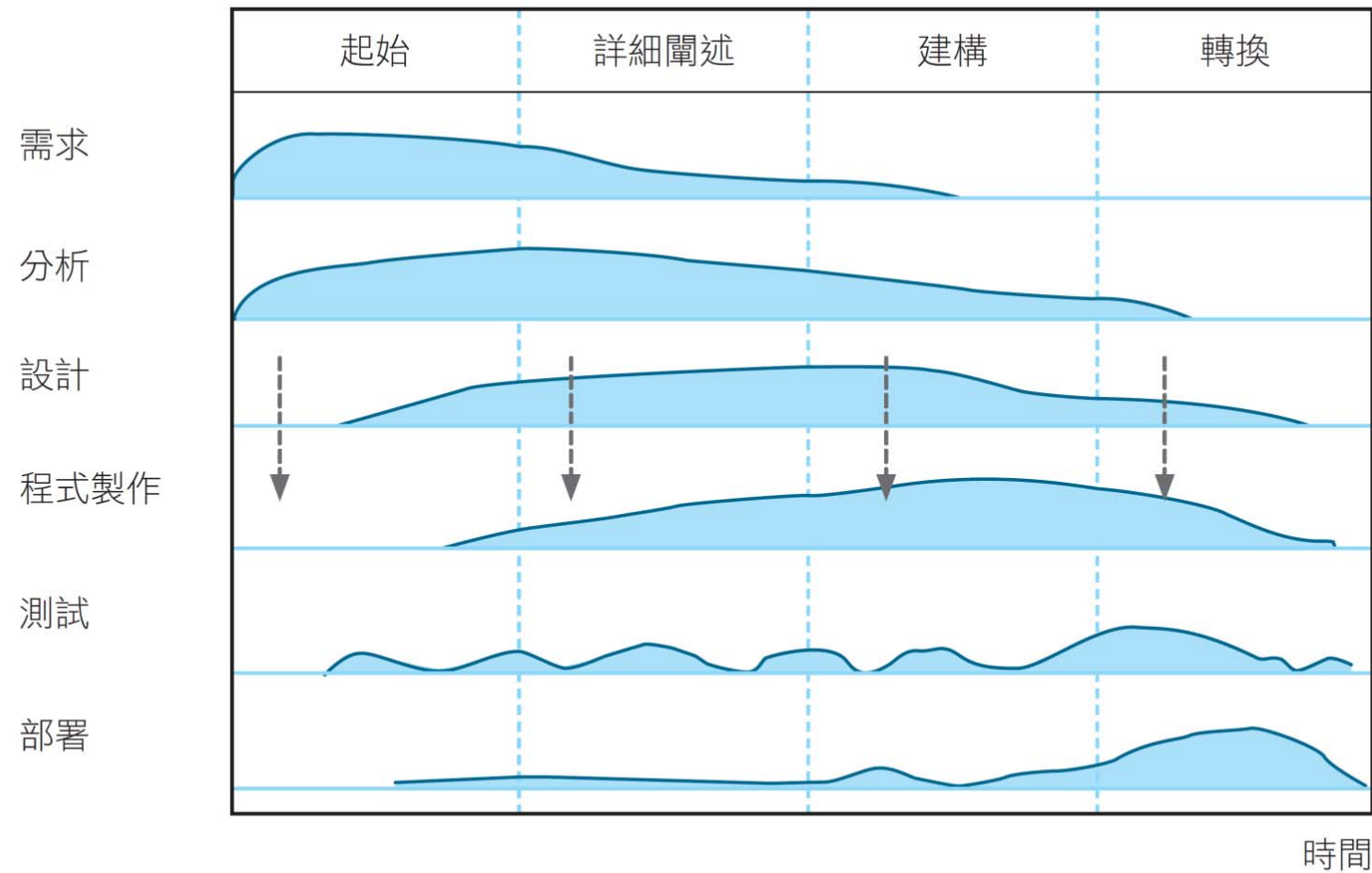


圖 1.5 統一程序之四個階段與軟體發展生命週期



# 統一程序要點

- 第一階段發展系統中最關鍵與風險高的部分，或者是技術上較為不確定的部分
- 兩個好處：
  - 一，經由後面階段的發展測試，能夠確保系統的關鍵功能正確執行或者發現問題及時改正
  - 二，如果技術太困難，發展小組也較有充分時間解決問題或者改變其他可行方案

# 制定程序所需要之階段與步驟

- 1. 規劃訪談問題與設計問卷。
- 2. 對主要使用者進行訪談，其他使用者以開會方式進行需求調查與資料收集。
- 3. 整理使用者需求，設定使用者目標。
- 4. 進行使用者目標驗證。
- 5. 編列事件表。
- 6. 管理者確認。

## 1-3 系統分析師與系統分析設計之工作

# 系統分析師的基本能力

- 資訊技術能力

- 分析設計的方法論
- 資料庫設計
- 作業系統
- 基本的程式語言
- 硬體與網路的運作原理

- 邏輯分析能力

- 所謂邏輯，即是由前面真實的事件或正確的敘述，能夠充分保證後面的事件或敘述是正確的
- 規劃出能因應環境與符合企業需求的系統

# 系統分析師的基本能力

- 企業的領域知識

- 瞭解現況趨勢
- 組織結構、管理技巧
- 企業的運作流程

- 人際溝通的技能

- 必須和不同使用者溝通以取得資訊和瞭解實際需求
- 傾聽、設身處地、引導與協助

# 系統分析師的基本能力

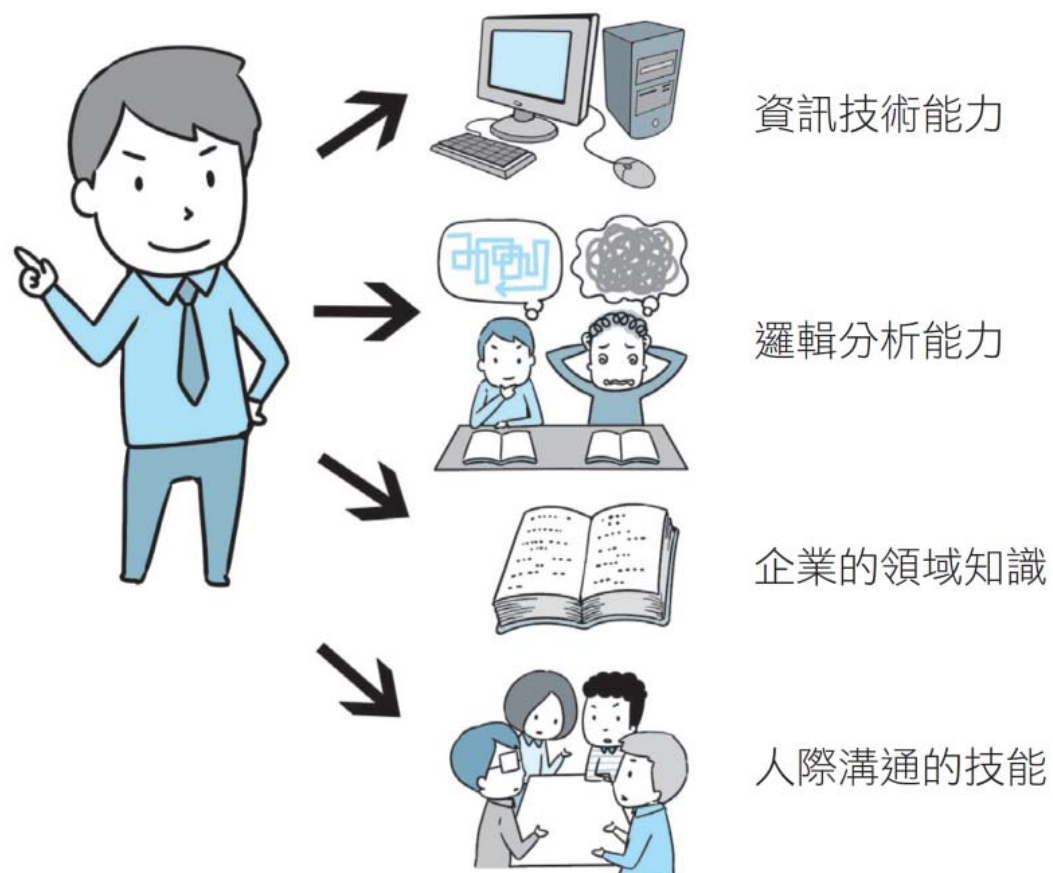


圖 1.6 系統分析師之能力

# 系統分析之工作

- 系統分析工作

- 需求調查
- 可行性研究
- 定義需求
- 撰寫規格書
- 擬定可行方案
- 管理者確認





# 系統規格書

- 系統分析工作的產出為系統規格書，系統規格書包含系統簡介、背景說明、使用者需求描述、環境說明、系統架構、系統規格、詞彙說明等，本書會在第 3 章作進一步說明。



# 系統設計之工作

- 系統設計工作

- 系統架構設計

- 系統功能設計

- 系統介面設計

- 資料庫設計

# 系統設計書

- 系統設計書包含：
  - 系統架構描述
  - 子系統功能描述
  - 個別類別或模組描述
  - 介面設計
  - 報表設計
  - 實作與測試計畫
  - 實際運作環境說明



# 總結

- ◆ 造成軟體問題的原因，在於使用者對於需求無法明確說明、環境變動太快等，要有效解決這部分的問題就需要做好系統分析設計的工作
- ◆ 目前最受歡迎與被認為最能提升品質與生產力的方法莫過於物件導向方法，而搭配物件導向方法最適當的程序即是統一程序。
- ◆ 物件導向方法將系統看成是不同的物件組成，每個物件中都擁有其自己的資料屬性和程序。