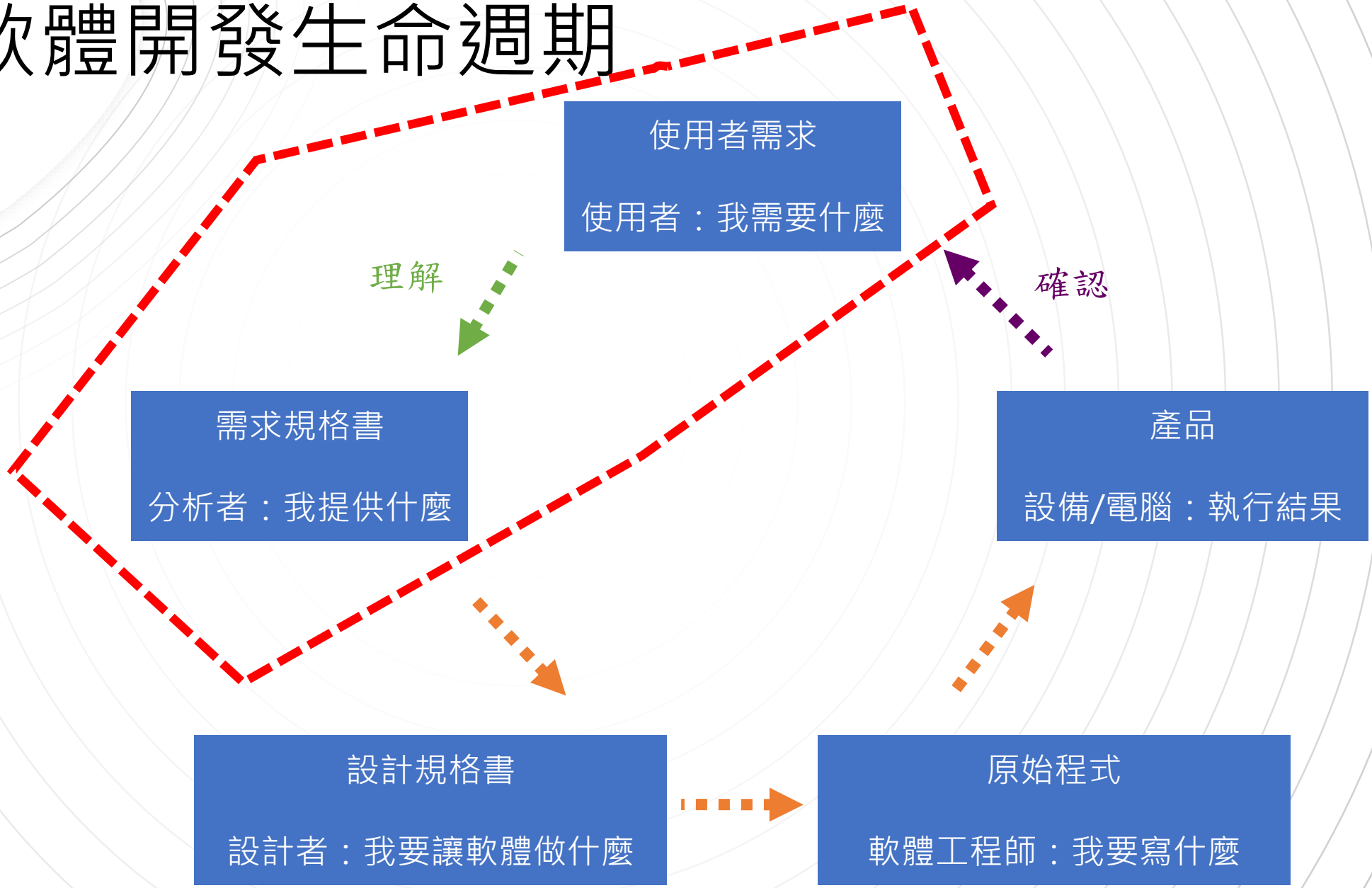
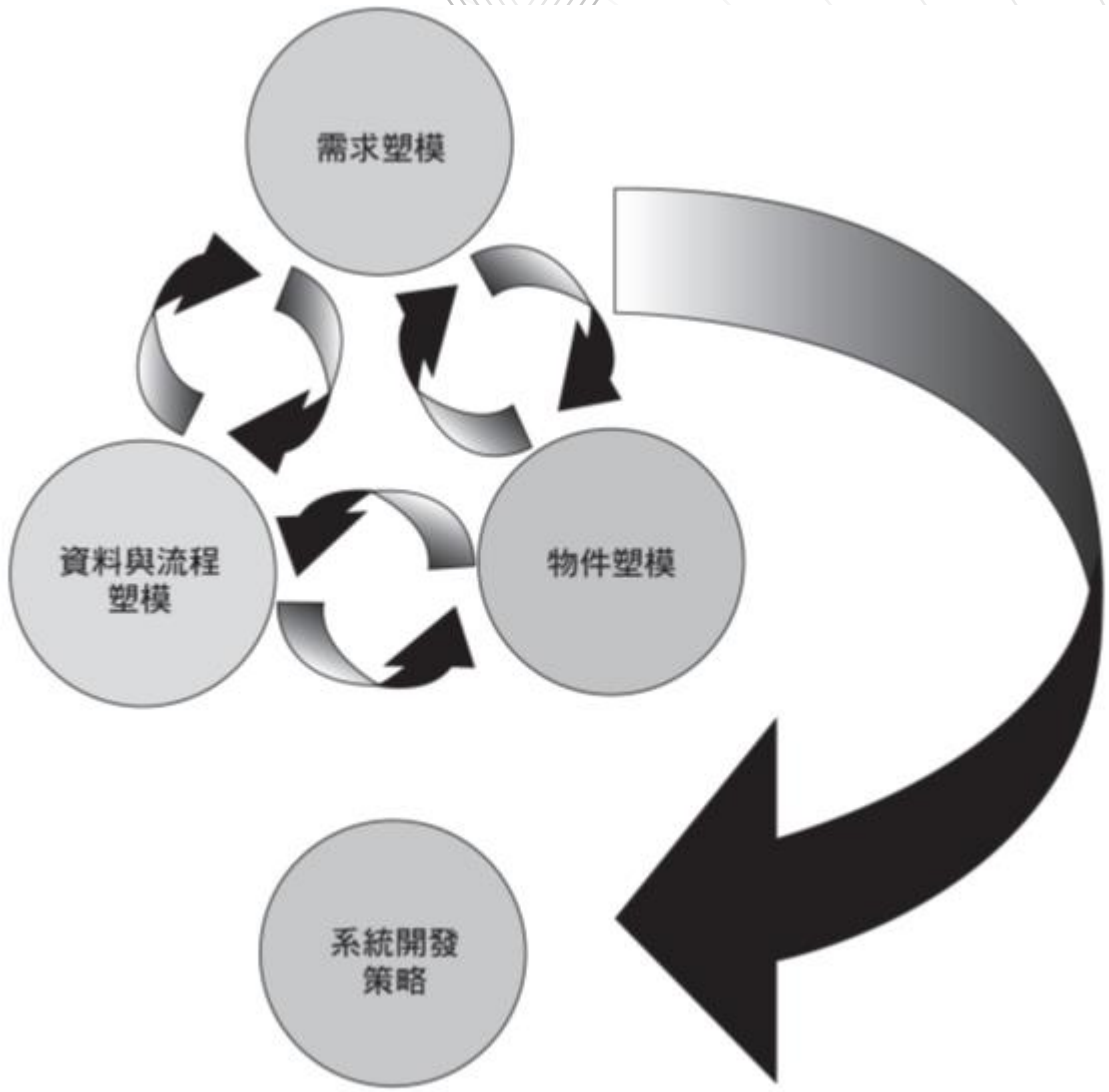
需求擷取與分析

軟體開發生命週期


使用者需求
使用者：我需要什麼
理解
確認
需求規格書
分析者：我提供什麼
產品
設備/電腦：執行結果
設計規格書
設計者：我要讓軟體做什麼
原始程式
軟體工程師：我要寫什麼

系統分析階段綜述

- 一般而言，系統分析的目的是了解提議中的計畫，確定它滿足商業需求，並且為系統開發建構紮實的基礎。
- 我們將運用塑模以及文件化的工具，以視覺化或文字來描繪提案的系統。
- 系統分析的**活動**
  - 需求塑模
    - **輸入(inputs)**
    - **輸出(outputs)**
    - **流程(process)**
    - **性能(performance)**
    - **安全性(security)**
  - 資料與流程塑模
  - 物件塑模
  - 開發策略
  - **系統需求文件 (system requirements document)**


需求塑模
資料與流程
塑模
物件塑模
系統開發
策略

需求擷取

- 分析階段的首要工作
  - 分析階段的首要工作在於需求的擷取與分析。
  - 根據許多的研究顯示，一個計畫之所以會失敗的原因之一常常是肇因於計畫初期，對於即將開發之系統所應提供的功能沒有切確的了解與掌握。基本上，一個無法滿足使用者需求的系統，不論所採用之資訊技術為何，其最終的命運均可想而知。
- 領域分析
  - 需求擷取可以從兩方面來著手。首先，你必須要從事領域分析(domain analysis)的工作。根據既有系統及其開發的歷史、領域專家的知識，背後的理論，辨別、收集以及組織相關資訊的過程。
  - 簡單地說就是也就是從各方面對目前既有的系統做資料蒐集。

需求擷取

- 蒐集範圍與種類
  - 在資料搜集的過程中，不要限定範圍或是種類，舉凡是跟領域問題有關聯的都要蒐集起來。任何資料都能讓你對於計劃領域有更深一層的了解。
  - 針對特定的需求，決定你要蒐集的方向，也就是跟計畫密切相關的資料。
- 領域專家
  - 領域分析是系統開發前的預備知識。對於待開發產品之背景資料分析與蒐集、對於產品領域知識與操作越了解，越能在將來做出好的決策。
  - 領域分析過程中，你會接觸到許多的人。對領域有深度了解的人稱為領域專家(Domain expert)，這些人大都是計畫的客戶、既有系統的使用者等等。他們對企業的各項活動以及業務流程最熟悉。而這些人也將在系統開發的分析階段扮演著重要的腳色。

需求擷取

- 了解領域知識的好處
  - 可以有效率的和客戶溝通
    - 一般來說，從事領域分析的大多是系統分析師。參予計畫的每一個成員都必須對相關的領域有基本的認識與了解。因為這可以增進開發團隊與客戶之間的溝通。
  - 了解企業既有的(As-is)流程活動，對於接下來即將(To-be)開發的系統會有實質的幫助。
    - 這一點，尤其對經常變動的企業邏輯 (business logic)或是商業規則(business rule)這方面尤其重要。
  - 預留未來新系統的開發空間
    - 從系統開發顧問公司的角度來看，在你為客戶開發系統的同時，你會對其業務有更深一層的了解。你甚至會了解到，客戶的許多既有的企業業務流程可以利用資訊系統的優點來改進，幫助企業改進執行的效率、工作生產力、降低成本開銷等等。對於這些觀察，可以經由與客戶高階執行階層討論與提出建議，以進而向客戶提出計畫書。為另一個新的計畫做準備。

擷取方式

- 了解了需求擷取的重要性，我們接著來看一下需求擷取有哪些方式。需求擷取的方式有很多，比較常用的方式如下
  - 企業既有的報表、表單、操作流程相關文件
  - 訪談
  - 小組討論
  - 腦力激盪

擷取方式

- 既有的報表、表單
  - 對於任何與開發中計畫相關之報表，表單之搜集。這些資料的主要來源為企業內既有的文件、業務流程說明、工作內容、作業描述等等。仔細閱讀這些資料以了解企業的核心流程，企業功能以及企業規則，幫助自己建立對該企業的一個普遍認識。對於人工作業時所採用之單據更需要仔細蒐集和保留。

訂購單　　公司名稱

客戶		電話				
訂貨日期		交貨日	年　月　日			
住址						
產品編號	名稱	規格	數量	單價	金額	備註
總計新台幣：　拾　萬　仟　佰　拾　元整　NT$：						
訂金：		刷卡：☐　卡號： 現金：☐				
餘款：						
1. 2. 3. 4. 5.						
經銷商： 代表號：(02) 123-4567						
業務員：　經手人：　送貨員：						

擷取方式(1)

- 訪談
  - 訪談是需求擷取方法中相當常用的一種方式。
  - 透過訪談你可以跟使用者有面對面討論與溝通機會，對於使用者的需求能夠有較真實的體驗。
  - 訪談之前必須要有詳細規劃，訪談之後必須將其結果詳實地紀錄下來

擷取方式(1)

- 訪談
  - 第一步：決定訪談對象
  - 第二步：建立訪談目的
    - 決定訪談的範圍
    - 確定需了解哪些事項
  - 第三步：設計訪談問題
    - 對訪談的問題進行標準化可以讓訪談有所依循，也避免節外生枝。
    - 須避免所謂的導引式問題 (leading questions)
    - 開放式問題(open-ended questions)
      - 專家訪談
    - 封閉式問題(closed-ended questions)：封閉式問題有很多種形式，包括：選擇題、下拉式選單、核取方塊和排名問題
      - 若目標受眾對您的調查主題並非特別感興趣
      - 若您需要收集可量化的數據
      - 若您需要對受訪者進行分類
    - 範圍式問題(range-of-response questions)

擷取方式(1)

- 訪談
  - 第四步：準備訪談
    - 訪談是很重要的會晤，不是閒聊，因此準備工作十分重要。
    - 最好限於一個小時之內完成。
    - 在訪談數天前必須將訪談主題寄給受訪者。
    - 如果想要查看資料，也應該告知受訪者事前備妥。
  - 第五步：執行訪談
    - 在訪談時先做自我介紹、描述專案內容，並解釋訪談目的。
    - **專注傾聽 (engaged listening)**。
    - 給對方足夠時間思考。
    - 在所有問題問完之後，應該彙整訪談重點，並解釋接下來會做什麼。例如，寄送下次的訪談通知。
    - 訪談結束時，要感謝對方，並鼓勵對方如果有新的資訊或任何問題都與我們聯繫，也可以詢問對方是否有其他主題可以討論。

擷取方式(1)

- 訪談
  - 第六步：記錄訪談
    - 最好盡量不要記筆記，筆記只做為簡單提示，盡量目視對方。
    - 在做完訪談之後，就必須馬上做記錄，並且評析所蒐集到的資訊。
    - 訪談之後可以寄感謝卡，載明訪談時間、地點、目的與主題，這樣受訪者就可以提供其意見。
  - 第七步：評量訪談
    - 除了獲取資訊之外，訪談時也要注意訪談是否有所失當。

- 失敗的訪談，變成聊天沒有結論。讓訪談有個目標，這樣才不至於讓訪談變成聊天

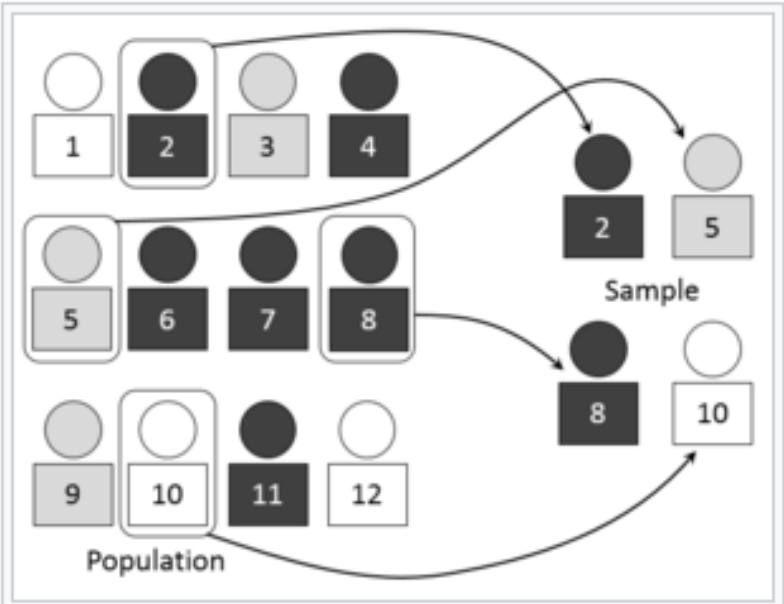
擷取方式(2)

- 小組定期開會討論
  - 定期開會討論與訪談的方式很類似。
  - 系統開發的過程中通常都會訂定有每週定期的開會時間表用以
    - 檢視計畫的進度
    - 確認工作細節
    - 確認需求的正確性
    - 各項相關任務的分派
    - 人力資源的調度等等事項。
  - 腦力激盪(brainstorming)
    - 針對特定問題、機會與議題的小組討論。
    - 此方法可以鼓勵新的構想、允許團隊參與、讓成員的想法與輸入彼此互為所用。

擷取方式(3)

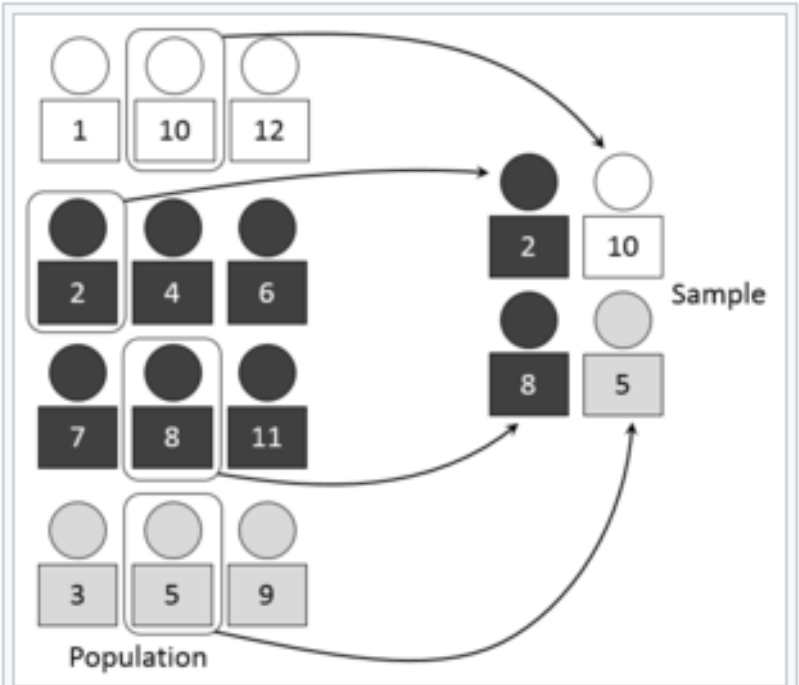
- 其他需求擷取的方式還包括有：
  - 觀察
    - 指實地觀察系統運作。它可以讓我們對系統有更進一步的了解。
    - 先準備好觀察項目以及所欲詢問的問題清單。
  - 問卷調查
    - 設計問卷最重要的是確定我們得到的資訊可以用做未來實情調查之用。
  - 抽樣調查：抽樣的主要目的是確保樣本能代表整個母體
    - 隨機抽樣 (random sample)
    - 系統抽樣 (systematic sample)
    - 分層抽樣 (stratified sample)

抽樣調查圖例說明



選擇簡單隨機樣本的示意圖

使用系統抽樣技術選擇隨機樣本的示意圖



使用分層抽樣技術選擇隨機樣本的示意圖

- 分層抽樣(stratified sampling)屬於隨機抽樣法(Radom sampling)中的一種，其方法為將抽樣母體分成性質不同或互斥的若干組，每一組為一個『層』(strata)，同層的性質要儘量相近，即變異要愈小愈好；不同層間的變異要愈大愈好，但分層組數不宜太多，可在6組以內。選擇分層的變數通常與研究的主題有直接的關聯，例如依BMI(身體質量指數)的大小將肥胖程度分為過瘦(18以下)、標準(18-23.9)、輕微過重(24-26.9)以及過重(27以上)等。其他常用的變項如性別、年齡、社經地位、都市化程度等

需求分析

- 利用各種需求擷取的方式，你獲得了許多跟即將開發的系統相關的資料。這些資料可能很凌亂，雜亂無章，因此你必須將這些資料做整理與分析。
- 需求的種類
  - 對於所擷取之需求資料可以分為兩大類：
    - 功能的需求 (functional requirement)
    - 非功能的需求(non-functional requirement)

需求分析

- 功能性需求
  - 功能的需求主要是在描述系統該做什麼。也就是系統要提供給使用者的服務項目。
  - 對於系統所提供之功能的描述可以包括什麼樣的輸入是這個功能所必須的、這個功能的處理流程與步驟、以及經過資料處理後，這個功能的輸出為何等等。

- 非功能性需求
  - 非功能的需求是指跟系統的執行效率，效能之需求，且可以量度的(measurable)的項目。

非功能性需求範例

- 反應時間(response time) ：對於使用者所觸發之事件的執行所將花費之時間。
- 使用性(usability)：描述對於一個正常的使用者所需之訓練時間。
- 可靠度(reliability)：可靠度可包含許多要項，最常見的是描述系統的失敗率。
- 效能(performance)：表現度可包含許多要項，最常見的是描述系統在每秒鐘可以處理的交易量。
- 維護性(maintainability)：描述任何可以增進系統維護之相關項目。比如說，編碼的準則，命名的標準等等。

需求分析

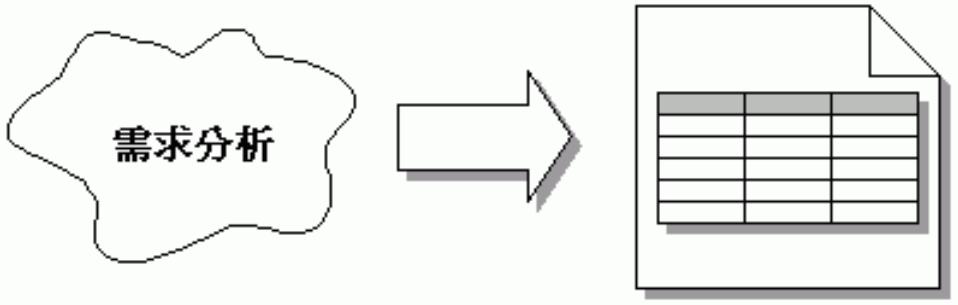
- 需求分析描述
  - 對於需求擷取所獲得的資料，你應該定義出所要解決的問題。定義問題的原則是：儘量使用淺顯易懂的句子，不要長篇大論，也不要太抽象，相關之流程敘述要記載明確。問題的敘述最好是從使用者的觀點出發。
- 例子
  - 對於一個電影院(火車、客運等等)訂票系統來說，我們可以將使用者的需求定義如下：
    ① 使用者必須登入以使用此系統。
    ② 使用者可以利用瀏覽器找尋相關電影並預訂電影票。
  - 如果你用“實現訂票流程自動化”來描述這個系統的功能，那就把問題的範圍定義變得太廣，很抽象。

- ATM例子
  - 大部分的人都使用過自動提款機(ATM)來從事提款，轉帳等事項。如果你仔細地觀察一下ATM的畫面，你會發現到一部ATM的主畫面以及其他可用的選項都會對應到一個事件上。比如說：提款，存款，餘額查詢，轉帳等等。這些不都是一部提款機所提供給使用者的功能嗎？一個使用者，必須製造某些事件來觸發或是告知系統你到底想要它做什麼。

需求分析

- 需求描述
  - 對於功能性需求的描述，你應該可以將它們歸納成事件。
  - 事件的寫法儘量用：
    - 主詞+動詞+受詞的形式。
  - 你可以把事件當作是需求描述的精簡版或是摘要(summary)。
- 事件表
  - 當你整理歸納出一大堆事件之後，可以將之紀錄於表格中，成為事件表。
  - 事件表是用來記錄系統功能很有用的工具。不要擔心功能到底需要怎麼被實現出來，先把系統當成一個黑盒子。這樣做的好處是讓參與計畫的人員能將焦點放在系統高層次的觀點，從外部來看系統，而不是系統內部的運作情況。在很多的經驗中我們發現人們一般都把焦點放在系統的How, 而不是系統的What。

需求分析



- 事件表欄位說明
  - 事件名稱：寫出造成系統去完成某事的事件。利用主詞+動詞+(受詞)的型式。
  - 觸發器(trigger)：引發事件的資料。
  - 來源：資料來源
  - 活動：事件發生時， 系統要執行的任務。活動要以動詞為開端。 例如：查詢 (Retrieve)...、更新(Update)...、產生(Create)...、取消(Delete)...。
  - 回應：什麼樣的輸出如果有的話？由哪裡產生？
  - 目的地：輸出到哪裡去。

事件名稱	觸發器	來源	活動	回應	目的地

需求分析

- 舉例說明
  - 一個線上購物系統，最明顯的事件就是”顧客下訂單”。我們可以分析出：事件名稱就是顧客下訂單。訂單是顧客下的，所以事件的來源來自於顧客。這個事件被觸發是因為訂單資料的到來，這是來自來源的一種要求。對於這個事件，系統應該要產生一筆新的訂單記錄，以記錄這個活動。所以，” 產生一筆訂單”是活動的名稱。而” 訂單編號”是這個活動的回應，此回應的目的地是使用者。

訂單產生之後要將訂單資料傳送到出貨部門

事件名稱	觸發器	來源	活動	回應	目的地
顧客下訂單	訂單	顧客	產生一筆新訂單	訂單編號 訂單	使用者 出貨部門

需求分析

- 事件圖思考方向
  - 嘗試著鑑別所有需要被儲存的資料。
  - 由建立(Create)、存取(Retrieve)、更新(Update)以及刪除(Delete)這四個動作來檢視系統可能發生的事件行為。這四個動作統稱為CRUD 。
  - 想一想有沒有以資料為導向的事件。
  - 想一想有沒有以時間為導向的事件。
  - 利用主詞+動詞+(受詞)的型式來歸納出事件。
  - 嘗試著鑑別各事件稍後會使用到的任何資訊。

軟體需求規格文件

- 軟體需求規格文件(Software Requirement Specification)記載著下列的項目
  - 系統目標
  - 系統範圍
  - 系統整體描述
  - 功能需求
  - 非功能需求
  - 系統所需提供之使用者、軟體、硬體等切面之需求
  - 其他資料

軟體需求規格書建議

IEEE 830 軟體需求規格書建議

- 簡介
  - 系統目的
  - 系統範圍
  - 名詞定義、縮寫
  - 參考
  - 系統概觀
- 系統整體性描述
  - 產品角度
  - 產品功能
  - 系統使用者
  - 系統限制
  - 系統假設
- 需求詳述說明
  - 外部介面需求
    - 使用者介面
    - 硬體介面
    - 軟體介面
    - 溝通介面
  - 功能性需求
    - 分類1（例如：子系統）
      - 功能性需求 R101
      - …
      - 功能性需求 R109
  - 非功能性需求
    - 效能、安全性、可靠性與可維護性
    - 交付、安裝與環境需求
    - 設計與實做限制
    - 測試需求與驗收標準
    - 技術/標準限制
    - 風險控管
- 附錄

需求規格書品質準則

- 明確且適當地陳述(Clearly and properly stated)
- 完整性 (Complete)
- 一致性 (Consistent)
- 能個別界定 (Uniquely Identified)
- 能適當地執行 (Appropriately implement)
- 能驗證 (Verifiable)

範例

一、前言

- 1.1 目的：本文件為軟體專案開發中之「軟體需求規格書」，目的在提供一可供參考依循的軟體需求分析作業程序，以利進行需求訂定作業並提昇分析品質，並可作為使用者、軟體需求分析人員、專案管理人員和廠商之間溝通的橋樑，亦是後續品保人員檢驗之依據
- 1.2 系統名稱：說明系統完整名稱
- 1.3 系統範圍說明：說明系統範圍、任務委託者、開發者、實際系統使用單位及本系統與其他系統互相關聯之訊息
- 1.4 縮寫說明
- 1.5 名詞定義
- 1.6 參考資料
- 1.7版本更新資訊：對於每次需求訪談與需求分析結果必須做成紀錄(附件一)，以做為

二、系統概述：在系統概述裡，分別敘述專案(系統)的來源及背景說明、使用者特點和系統目標等。

- 2.1專案來源及背景
- 2.2用戶特點：若有用戶特殊性值或使用條件，請於此處說明
- 2.3本系統設定目標
- 2.4 需求塑模方法：本文建議採用物件導向塑模方法論(OOA：物件導向分析、OOD 物件導向設計)來進行，而物件導向塑模工具則採用統一塑模語言2.0 版(UML2.0 ,Unified Modeling Language)來展現，其中主要使用：使用案例圖(Use CaseDiagram)、循序圖(Sequence Diagram)、活動圖(Activity Diagram)等圖形技術來描述及溝通需求

- 三、系統環境：此處根據需求書及議約內容陳述。
  - 3.1 開發工具與設備環境：描述系統所需的軟體及工具清單及版次
  - 3.2 系統運行網路環境：描述系統之軟硬體部署環境、與網路架構等
  - 3.3 系統運行硬體環境：本節描述系統所需硬體伺服器規格等級及數量等，並說明設備來源
  - 3.4 系統運行軟體環境：描述系統之軟體規格、與資料庫配置架構等
  - 3.5 系統限制

四、功能性需求：我們採用物件導向分析作為主要的系統塑模的方法，並使用UML 2.0 做為塑模語言。在UML 中，任何一個角度對系統所做的抽象定義，都可能需要幾種模型圖來描述，而這些來自不同角度的模型圖最終組合成整個系統。而在本需求書中最低要求使用：使用案例圖(Use Case Diagram)、循序圖(Sequence Diagram)、活動圖(ActivityDiagram)等圖形技術來描述及獲取需求。

五、對性能的要求：有時我們會用非功能性需求的說法描述本系統的性能，主要描述系統軟體的非功能性需求，如可靠性、資料正確性、安全需求、操作需求等。

- 5.1人機介面環境需求：詳細說明各界面之需求，以及界面傳輸資料之需求，並考慮使用者操作之便利性
- 5.2回應時間需求
- 5.3可靠性需求
- 5.4系統安全性需求
- 5.5需求規格回溯(圖)表

六、產品交付說明

七、確認

需求驗證：

① 需求的正確性：開發人員及用戶都須進行每個版本的複查工作，以確保將用戶的需求充分、正確的表達出來，應於專案開始之初，及邀請用戶全程參與需求的釐清任務。

② 需求的一致性：一致性指與業務需求需要一致，不可相矛盾，在開發前需要解決所有需求不一致的部份，驗證沒有任何衝突或含糊的需求。

③ 需求的完整性：需要驗證是否所有可能的狀態變化、轉入資料、產品功能和限制，都在需求中描述，不能遺漏任何必要的需求訊息。

④ 需求的可行性：驗證需求是否接確實可行，而且是在專案執行期間可以備實施的，並應有技術人員隨時檢視需求技術的可行性。

⑤ 需求的必要性：驗證需求是客戶需要的。每一條需求都應該把客戶真正需要的和最終系統所需遵從的標準記錄下來，「必要性」可以理解為為每項需求都是用來授權編寫文件的根源。要使每項需求都能回溯至某項客戶的輸入。

⑥ 需求的可檢驗性：驗證是否能寫出測試案例(Test Case)，來滿足需求驗證。撰寫測試案例和其他自動化測試工具來驗證需求。

⑦ 需求的可追蹤性：驗證需求是否可以追蹤，應能在每項軟體需求與他的來源和設計元素、原始程式碼、測試案例之間建立連接，這種可追性要求每項需求以一種結構化的方式編寫文件並單獨標明，而不是大段大段的敘述。

⑧ 使用者簽認：每個版本的需求書產出需要參與者簽字確認。

需求變更管理：

① 確定變更控製流程：確定一個選擇、分析和決策需求變更的過程，所有需求變更都須遵循此流程，此流程須含軟體變更管理組織(SCCB, Software ChangeControl Board)之確認程序。

② 進行變更影響評估：評估需求變更對專案進度、資源、工作量和專案範圍以及其他需求的影響。

③ 跟蹤變更影響的產品：當進行某項需求變更時，需要找到與此變更相關的其他需求、設計文件、原始程式碼、測試案例、這些相關部分可能也要修改。

④ 建立基準和控制版本：為需求文件建立一個基準(BaseLine)，這是一致性需求在特定時刻的一個凍結，之後的需求變更就遵循變更控制流程即可。

⑤ 維護變更歷史資料：紀錄變更版本日期及所作的變更、原因，還包括由誰負責更新和更新的新版本號等情況。

⑥ 跟蹤每項需求的狀態：這裡狀態包括：「確定」、「已實現」、「暫緩」、「新增」、「變更」等，可以用資料紀錄的方式來記錄一項需求。

THANKS.