第5章 需求工程

- 5-1 需求工程的基礎
- 5-2 需求擷取
- 5-3 定義需求
- 5-4 需求規格
- 5-5 需求驗證

5-1 需求工程的基礎

- 5-1-1 需求
- 5-1-2 需求工程
- 5-1-3 需求的作業流程

5-1-1 需求-說明

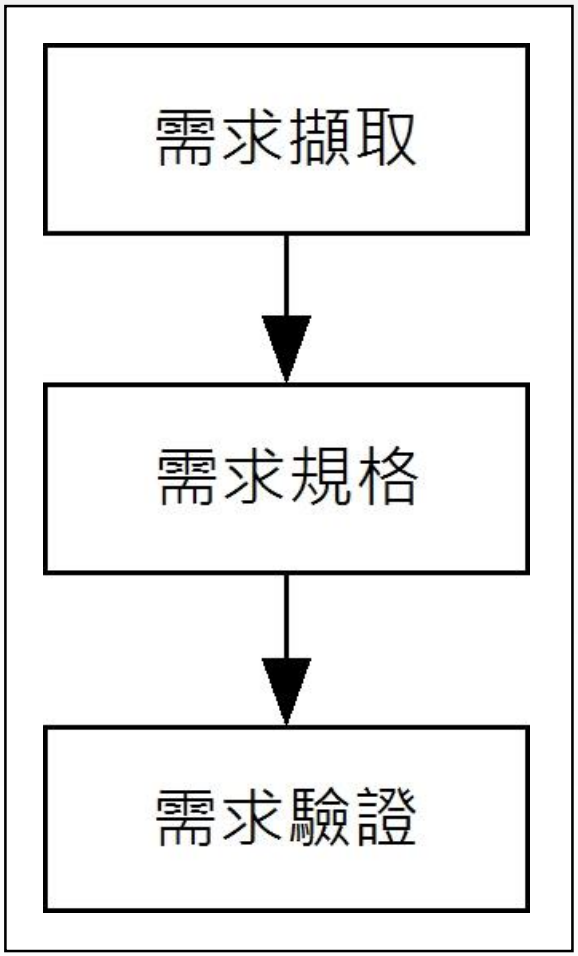
- 「需求」（Requirements）是使用簡單、高階和抽象的文字敘述來描述使用者需要的系統服務和操作限制，或正式定義系統詳細功能的規格書（Davis，1993）。
- 需求包含系統特點的描述、系統如何運作的規格和限制，一般來說，需求是在描述系統應該作什麼（What），而不是系統如何作到這些功能（How）。

5-1-1 需求-需求等級

- 基本上，需求可以分成兩種等級，如下所示：
  - 使用者需求（User Requirements）：使用人類語言加上圖形來描述系統提供的功能和操作限制，這是從使用者角度描述的需求，在第8章建立的使用案例模型就屬於此類需求。
  - 系統需求（System Requirements）：詳細描述系統功能、服務和操作限制的系統規格文件或稱為功能規格（Functional Specification），需要明確定義實作的系統，這是從系統開發者角度描述的需求。

5-1-2 需求工程-說明

- 「需求工程」（Requirement Engineering，RE）是一門學科來找出、分析、文件記錄和驗證系統功能和操作限制，簡單的說，就是在找出客戶與使用者的真正需求，可以分成三個階段，如右圖所示：



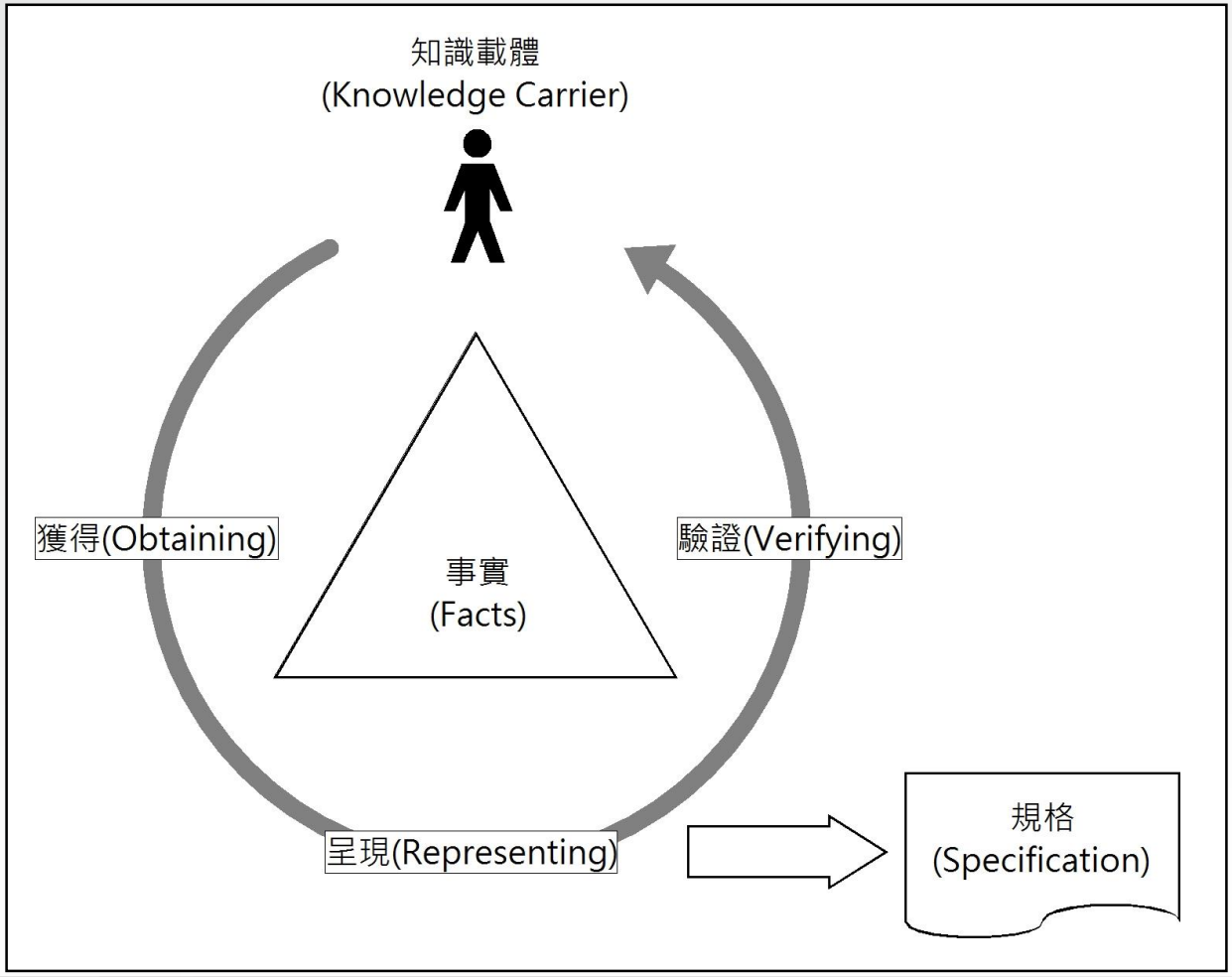
5-1-2 需求工程-各階段的說明

- 需求擷取（Requirements Elicitation）：實際從系統的使用者、客戶或利益相關者取得系統需求，也稱為需求收集（Requirements Gathering）。
- 需求規格（Requirements Specification）：使用適當語言和符號描述取得的需求，也稱為需求文件（Requirements Document），可以描述為什麼需要此系統，最後完成開發的系統樣貌，其中有很大部分是正式需求清單的列表。
- 需求驗證（Requirements Validation）：針對需求規格來確認文件內容的完整、一致和正確性。

5-1-3 需求的作業流程-找出和發現事實

- 需求的作業流程是一個找出和發現事實（Facts）的過程，也就是明確定義出解決的問題是什麼，使用者的真正需求為何？我們需要從知識載體（Knowledge Carrier），即了解系統的人，例如：客戶、使用者或領域專家（Domain Experts）來獲得（Obtaining）、呈現（Representing）和驗證（Verifying）事實。

5-1-3 需求的作業流程-圖例



5-1-3 需求的作業流程-圖例說明

- 需求的作業流程可以產生需求規格（Specification），其簡單說明如下所示：
  - 獲得（Obtaining）：從領域專家等知識載體取得領域知識（Domain Knowledge）來獲得事實（Facts），以便找出使用者真正的需求是什麼，即需求擷取。
  - 呈現（Representing）：使用文字描述或圖形來呈現事實（Facts），即需求規格。
  - 驗證（Verifying）：我們需要知識載體的客戶、使用者或領域專家等來驗證我們發現的事實（Facts）是否正確，即驗證需求。

5-2 需求擷取

- 5-2-1 領域分析與領域知識
- 5-2-2 識別利益相關者
- 5-2-3 需求擷取活動
- 5-2-4 記錄需求

5-2 需求擷取

- 需求擷取（Requirement Elicitation）就是在調查系統的需求，我們需要了解領域知識和識別利益相關者（Stakeholders），這是需求擷取前的準備工作，然後進行需求擷取活動來找出和發現需求。

5-2-1 領域分析與領域知識-領域分析

- 軟體工程的「領域分析」（Domain Analysis）或稱為產品線分析（Product Line Analysis）是一種收集、分析和組織軟體系統相關背景知識的過程，其主要目的是讓系統開發者學習目標軟體系統的相關背景知識。

5-2-1 領域分析與領域知識-領域知識

- 領域分析的主要目的是取得領域知識（Domain Knowledge），領域知識是目標軟體系統操作環境的所有背景知識，例如：金融機構的資訊系統，我們需要蒐集金融相關知識；醫院的資訊系統是醫療相關知識等。
- 為了取得領域知識，我們可以從相關書籍、現有系統和相關文件來取得，另一種方式是請教領域專家（Domain Experts），他們是一些對此行業有深入了解的人，大部分就是客戶或使用者。

5-2-2 識別利益相關者-說明

- 「利益相關者」（Stakeholders）是一些對系統有興趣的人，他們有可能直接或間接影響到軟體系統的設計與開發成敗，因為這些人大部分就是客戶或使用者，換句話說，我們需要在公司或組織中找出所有利益相關者，以便在進行需求擷取活動時，可以完整找出系統需求。

5-2-2 識別利益相關者-種類

- 系統開發常見的利益相關者種類，其說明如下所示：
  - 執行者（Executive）：他們是系統開發專案付款的人，通常是公司、部門主管或實際執行商業決策的一些人。基本上，他們不會涉入軟體開發的技術層面，只在商業流程和規則方面提供意見，他們的經驗和專業與使用者有很大的不同。
  - 終端使用者（End User）：實際使用軟體系統的使用者，沒有人比他們（包括你）更了解需要建立的系統是什麼？如果沒有特別說明，在本書的使用者就是指終端使用者。
  - 專家（Expert）：有時我們需要請教公司顧問，或其他部門的員工，例如：業務、支援、法務和會計人員，他們可以提供系統一些非功能上的需求資訊。

5-2-2 識別利益相關者-如何識別

- 在實務上，我們有一些方法可以在公司或組織中找出哪些人員是系統開發專案的利益相關者，如下所示：
  - 是誰在操作和維護系統。
  - 是誰會因為系統建立，可以提供他們工作上的協助，也就是因為系統而受益的人。
  - 是誰付款、決定預算和主導系統開發。
  - 有誰反對新系統的開發。

5-2-3 需求擷取活動-背景閱讀

- 背景閱讀（Background Reading）也稱為背景研究（Background Research），對於外包軟體廠商的開發人員來說，需求擷取的第一步就是瞭解公司或組織的背景，可以協助開發人員與利益相關者之後的訪談或會議，我們可以從公司或組織內部的相關文件來進行背景閱讀。

5-2-3 需求擷取活動-訪談

- 訪談（Interviewing）是最廣泛使用的需求擷取活動，這是一個直接和利益相關者面對面進行溝通的機會，不過，它也是一種需要技巧和敏感度才能進行的活動。
- 訪談活動需要事前規劃，我們需要確認訪談對象、目的、準備好訪談問題、列出需參考文件和簡單的行程表，並且在訪談之後詳細記錄訪談的結果。

5-2-3 需求擷取活動-問卷調查

- 問卷調查（Questionnaires）不能取代訪談，它只是在時間和地點不允許且無法配合時採取的替代方案，我們可以透過問卷形式來替代訪談。不過，如果你完全不進行訪談，就會發現準備了一大堆問題，但是，最後仍然無法得到真正的答案。
- 問卷調查的主要目的是補訪談的不足，對於訪談時不能觸及的關鍵或敏感問題，就可以設計成問卷形式來解決，不只如此，因為問卷是一種可以更廣泛取得受調查者需求的方式，當時間上不允許時，我們可以針對主要受訪者進行訪談，其他就使用問卷調查。

5-2-3 需求擷取活動-觀察

- 觀察（Observation）是需要花費大量時間，卻很有價值的需求擷取活動，系統開發人員直接到使用者的工作場所，現場觀察使用者每天的工作情況，以便進一步了解工作內容，如此，可以獲得比訪談更有用的資訊。
- 因為訪談只能了解正常的工作情況和內容，並無法得知一些例外狀況，或使用者自己沒有注意而忽略的部分，例如：一些現存系統之外的人工步驟，就可能在訪談時被忽略掉。

5-2-3 需求擷取活動-腦力激盪會議

- 召開腦力激盪會議（Brainstorming Meeting）是另一種需求擷取的好方法，我們可以和利益相關者一起參與會議來確定一些關鍵性的需求。
- 腦力激盪可以集思廣義，從參與者的大腦中激發出所有可能的結果。

5-2-4 記錄需求

- 在進行需求擷取活動時，我們需要將取得的需求記錄下來，通常是使用一個統一表格來記錄，因為一致的表格內容，可以幫助我們定義、分類和管理需求。在表格中需要記錄的基本資訊，如下所示：
  - 一個簡短的標題。
  - 詳細描述需求的內容。
  - 是誰記錄此需求。
  - 需求來源是哪一種需求擷取活動。
  - 一個理由，為什麼此需求是必須的？
  - 影響此需求的利益相關者清單。
  - 依據需求來源設定其可能的優先等級。

5-3 定義需求

- 5-3-1 需求描述
- 5-3-2 需求分類
- 5-3-3 需求屬性
- 5-3-4 需求整理
- 5-3-5 事件分割法的事件表

5-3 定義需求

- 在取得原始需求資料後，就可以定義需求，其主要工作有：描述需求、將需求分類、指定需求屬性和整理需求。事實上，定義需求就是在進行需求分析（Requirement Analysis）。
- 請注意！第5-3-1節到第5-3-4節屬於「傳統」的需求分析方法，在第8章的使用案例模型是一種物件導向的需求分析方法。第5-3-5節的事件分割法是一種適用在傳統和物件導向的系統分析技術，其建立的事件表，可以幫助我們在第8章找出動作者和使用案例。

5-3-1 需求描述-說明

- 需求描述是將需求擷取得到的原始需求資料，重新定義成我們需要解決問題的需求，描述需求應該儘量使用簡單和淺顯易懂的句子來進行描述，句子不需要太長，更不可長篇大論，我們應該從客戶和使用者的觀點來描述需求。

5-3-1 需求描述-原始的需求描述

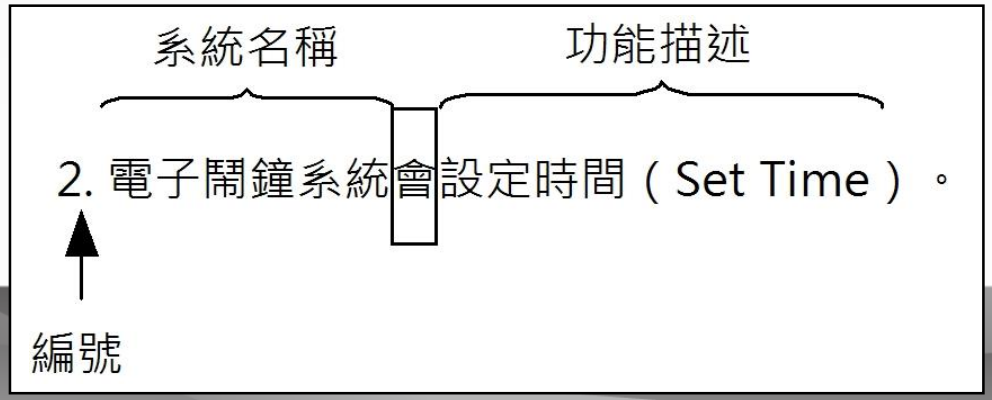
- 例如：線上訂購系統的需求描述，如下所示：
  - 客戶可以加入會員。
  - 會員可以登入系統。
  - 客戶可以瀏覽商品目錄。
  - 客戶可以瀏覽商品明細。
  - 客戶可以新增訂購商品。
  - 客戶可以刪除訂購商品。
  - 客戶可以更改訂購商品數量。
  - 會員可以下訂單（即結帳）。

5-3-1 需求描述-語法

- 在實務上，建議使用一個簡單且標準的語法來描述需求，如下所示：

  編號. 系統名稱「會 | 會提供」功能描述

- 上述語法是以數字編號開始，之後是系統名稱，然後使用「會」或「會提供」等表示系統擁有的功能，最後是功能描述，例如：電子鬧鐘系統的一個需求描述，如下所示：



5-3-1 需求描述-標準語法的需求描述

- 現在，我們可以將之前線上訂購系統的需求改寫成標準語法來描述，如下所示：
  1. 線上訂購系統會提供客戶加入會員。
  2. 線上訂購系統會提供會員登入系統。
  3. 線上訂購系統會提供客戶瀏覽商品目錄。
  4. 線上訂購系統會提供客戶瀏覽商品明細。
  5. 線上訂購系統會提供客戶新增訂購商品。
  6. 線上訂購系統會提供客戶刪除訂購商品。
  7. 線上訂購系統會提供客戶更改訂購商品數量。
  8. 線上訂購系統會提供會員下訂單。

  ……..

5-3-2 需求分類-功能性需求(說明)

- 功能性需求是在描述系統應該做什麼和期望做什麼（What），它是一段系統功能或服務的描述，和當特殊資料輸入時，系統需要如何反應，或發生特殊情況時，系統所做的行為。
- 對於一些特殊案例，功能性需求還可能需要明確描述系統不能執行哪些功能。

5-3-2 需求分類-功能性需求(範例)

- 當我們進行需求擷取活動時，回答內容出現「系統必須…」、「系統應該…」或「系統可以…」等句型時，就表示句子中隱含功能性需求的服務或功能，英文是must、shall或should。例如：ATM自動提款機的功能性需求，如下所示：
  1. ATM自動提款機系統會檢查金融卡是否是一張有效卡片 。
  2. ATM自動提款機系統會驗證客戶輸入的密碼是否正確。
  3. ATM自動提款機系統會提供客戶提款。
  4. ATM自動提款機系統會提供客戶存款。
  5. ATM自動提款機系統會提供客戶查詢帳戶餘額。
  6. ATM自動提款機系統會提供客戶轉帳。

5-3-2 需求分類-非功能性需求(說明)

非功能性需求	說明
效能（Performance）	系統可以滿足客戶的最低執行效能，通常會包含多個項目來進行評斷，例如：每秒處理的交易量和反應時間（Response Time）等，反應時間是指必須在一定時間內得到回應
使用性（Usability）	系統是否容易使用，即使用者可以正常上線的訓練時間
可靠性（Reliability）	系統是否可以持續可用，反過來，就是允許系統失效的平均時間或頻率，例如：一個月不可超過一次系統失效
操作性（Operational）	系統可以連續操作的天數，例如：連續一個月或365天
維護性（Maintainability）	系統維護的相關事項，確保可以應付未來的需求，例如：下一個版本需要的修正
保密性（Security）	關於保密和敏感資訊的處理，例如：不同權限的帳戶管理和加密
財務考量（Financial）	專案成本考量的預算上限，通常不會列在需求規格文件，而是在合約書或開發計劃
合法性（Legal）	系統環境操作的合法性，簡單的說，我們是否可以合法使用此系統

5-3-2 需求分類-非功能性需求(範例)

- 一般來說，我們談到的需求都是指功能性需求，但是，非功能性需求一樣十分重要，如果沒有滿足非功能性需求，意味著整個系統都可能無法使用。例如：ATM自動提款機的非功能性需求，如下所示：
  1. ATM自動提款機系統會使用128位元的加密方式進行連線。
  2. ATM自動提款機系統會在4秒之內完成金融卡的驗證。
  3. ATM自動提款機系統會在3秒之內完成密碼的驗證。
  4. ATM自動提款機系統會在20秒內完成轉帳。

5-3-3 需求屬性-說明

- 需求屬性是需求的描述資料（Meta-Data），每一個需求都需要使用一組屬性來記錄需求的額外資訊，例如：完成日期和優先等級等
- 基本上，需求屬性的語法是由屬性名稱和值組成，例如：名為【優先等級】的屬性，其值為MoSCoW規則M。

5-3-3 需求屬性-MoSCoW規則的優先等級

- 需求屬性最常使用的是需求的優先等級（Priority），我們常常使用MoSCoW規則來指定需求的優先等級，如下表所示：

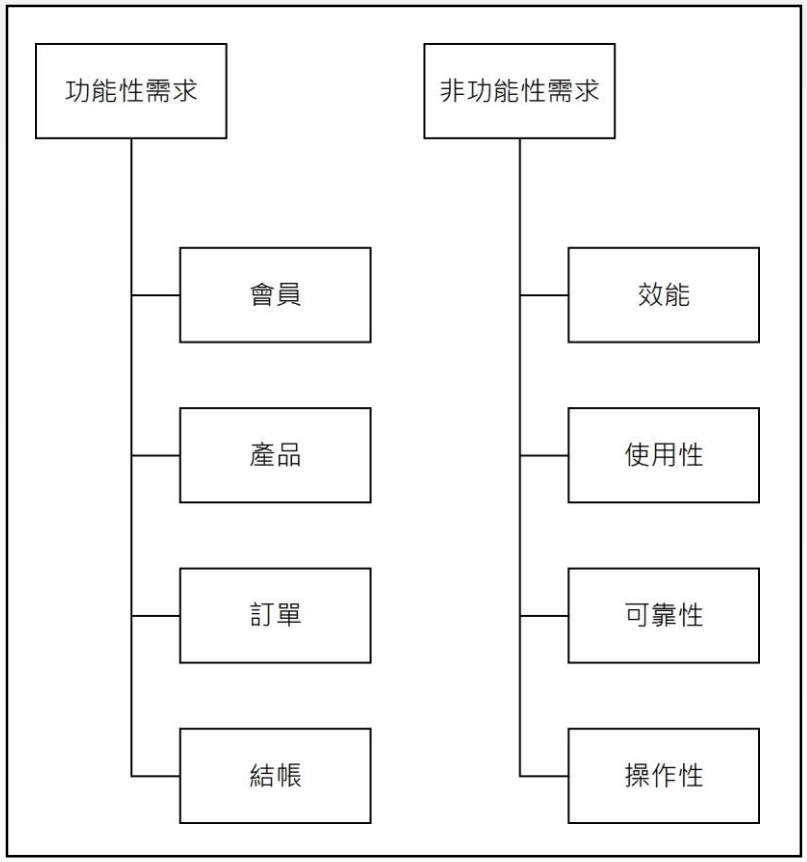
MoSCoW規則	說明	屬性值
必須有（Must-haves）	一定必須要有的需求	M
應該有（Should-haves）	需求很重要，但刪除也不會影響專案的開發	S
可能有（Could-haves）	可選擇是否開發的需求	C
期望有（Want-haves）	等到系統開發較成熟後，再考量是否開發的需求	W

5-3-3 需求屬性-Rational統一流程定義的需求屬性

需求屬性	說明
狀態（State）	需求的狀態，其屬性值如下所示： ☐ 建議：需求只是建議，未經討論和同意 ☐ 允許：需求允許開發 ☐ 拒絕：需求拒絕開發 ☐ 完成：需求已經開發完成
效益（Benefit）	需求的重要性，其屬性值如下所示： ☐ 嚴重的：有嚴重問題的需求，需求已經開發完成，但是利益相關者不認可此需求 ☐ 重要的：重要的需求，刪除需求會影響系統的滿意度 ☐ 有用的：有用的需求，刪除需求會影響系統的接受度
成本（Effort）	需求完成所需的時間和資源
風險（Risk）	風險等級是高、中或低
穩定性（Stability）	需求變動的可能性是高、中或低
目標釋出版本（Target Release）	預計需求完成的產品釋出版本

5-3-4 需求整理

- 對於大型專案的資訊系統開發來說，因為需求描述的數量很龐大，通常超過100個以上，此時，我們需要將需求分類（Taxonomy）管理，也就是使用階層架構來管理需求，幫助我們執行需求的相關工作。
- 需求分類的第一步是分類成：功能性和非功能性需求，然後再細分出下一層的不同分類，如右圖所示：



5-3-5 事件分割法的事件表-說明

- 「事件分割法」（Event Partitioning）是一種使用在傳統或物件導向的系統分析技術（Systems Analysis Techniques），在傳統的系統分析方法是用來識別活動；物件導向是識別「使用案例」（Use Case）。
- 使用案例是系統回應使用者請求所執行的活動，可以使用事件分割法來辨識出使用案例，在第8章我們也會使用此技術來找出使用案例，這一節主要是說明如何使用事件分割法的事件表（Event Table）來定義系統需求。

5-3-5 事件分割法的事件表-事件的基礎(說明)

- 軟體系統的操作過程就是一個與使用者互動的過程，系統需要等待使用者下達指令，例如：選擇選項或按下按鈕（即產生事件），系統才會回應請求來執行特定的功能，簡單的說，系統功能不會沒有任何原因而自動的執行，它一定是因為產生事件才會執行。
- 換句話說，系統一定需要等到某位使用者的特定操作或特定情況觸發事件後，系統才會回應事件來執行指定的功能。因為系統功能與事件之間有因果關係，所以，我們可以從事件角度來描述系統需求。

5-3-5 事件分割法的事件表-事件的基礎(種類)

- 基本上，事件可以分為三種，其說明如下所示：
  - 外部事件（External Event）：發生在系統之外的事件，主要是因為使用者需要執行交易、輸入資料、取得資訊和更新資訊等操作產生的事件，例如：客戶輸入使用者名稱和密碼，產生使用者登入系統的事件等。
  - 時間點事件（Temporal Event）：在系統內特定時間點到達時觸發的事件，通常是一些輸出報表產生的事件，例如：系統在特定時間點產生管理報表、交易報表、帳單和催繳單等。
  - 狀態事件（State Event）：系統內部狀態改變所觸發的事件，例如：管理者關閉系統等。

5-3-5 事件分割法的事件表-事件名稱的語法

- 事件名稱的基本語法，如下所示：

  主詞+動詞+受詞

- 上述語法可以定義事件名稱，例如：客戶新增訂購商品、會員登入系統、客戶註冊會員和會員更改會員資料等。

5-3-5 事件分割法的事件表-事件表(欄位說明)

- 事件表（Event Table）就是使用表格列出事件來描述系統需求。事件表欄位說明，如下表所示：

欄位	說明
事件	事件名稱，這是造成系統去完成某些工作的事件
觸發器	產生事件的原因，這是系統如何知道事件發生的原因，例如：指定操作、輸入資料或時間到了等情況
來源	事件來源的外部實體，通常是系統的使用者或外部的其他系統
活動/使用案例	當事件產生後，系統回應事件的操作，也就是系統會做什麼事？
回應	系統回應事件後產生的輸出，它是由系統產生的輸出結果（如果有的話）
目的地	輸出結果的目的地，即誰會取得輸出結果

5-3-5 事件分割法的事件表-事件表(範例1)

- 範例一：外部事件的會員登入系統事件表，如下表所示：

事件	觸發器	來源	活動/使用案例	回應	目的地
會員登入系統	會員輸入使用者名稱和密碼	會員	登入系統	成功登入或登入失敗	會員

- 範例二：外部事件的客戶下訂單事件表，如下表所示：

事件	觸發器	來源	活動/使用案例	回應	目的地
客戶下訂單	建立新訂單	客戶	產生新訂單	訂單明細與訂單確認	客戶與出貨

5-3-5 事件分割法的事件表-事件表(範例2)

- 範例三：時間點事件的產生交易報表事件表，如下表所示：

事件	觸發器	來源	活動/使用案例	回應	目的地
產生交易報表時間	下班前		產生交易報表	交易報表	會計部門

5-4 需求規格

- 5-4-1 需求規格的基礎
- 5-4-2 需求規格文件

5-4-1 需求規格的基礎-內容

- 需求規格（Requirements Specification）就是使用適當語言和符號來描述我們取得的需求，不過，需求規格並沒有一定的寫法，我們可以自行定義需求規格的內容和撰寫方式。
- 在需求規格除了列出需求清單（功能性與非功能性需求）外，我們還需要說明為什麼需要這套系統，和最後完成開發的系統樣貌。
- 基本上，需求規格需要描述使用的開發工具、專案預算、成員簡介和開發所需的預估時間表，最後開發完成的成品如何送交和安裝等可能影響專案開發的相關資訊。

5-4-2 需求規格文件

- 軟體系統的需求規格文件並沒有固定和制式規格，各家軟體廠商都有專屬格式，其內容也大不相同。筆者以著名的IEEE/ANSI 830-1998（IEEE，1998）標準需求規格架構為例，如下所示：

1. 簡介（Instruction）
    - 1.1 需求規格的目標
    - 1.2 系統範圍
    - 1.3 定義與縮寫
    - 1.4 參考
    - 1.5 文件其他部分概觀
2. 一般描述（General Description）
    - 2.1 系統架構
    - 2.2 系統功能
    - 2.3 使用者特性
    - 2.4 一般性限制
    - 2.5 相關軟硬體
3. 特殊需求（Specific Requirements）
    - 3.1 功能性需求

3.2 非功能性需求

4. 附錄（Appendices）
5. 索引（Index）

5-5 需求驗證

- 5-5-1 需求驗證的基礎
- 5-5-2 需求驗證技術

5-5-1 需求驗證的基礎

- 需求驗證的目的是確認需求規格（Requirements Specification）的內容是正確的，而且能夠正確反應客戶或使用者的需求。簡單的說，需求驗證是在獲得所有利益相關者、開發者、客戶和使用者都同意可以解決需求問題。主要在回答三個問題，如下所示：
  - 需求規格是否完整？
  - 需求規格是否一致？
  - 需求規格是否正確？

5-5-2 需求驗證技術-手工檢查

- 需求驗證最原始的方式是使用手工檢查（Hand Checking），也就是讓利益相關者閱讀需求規格來驗證規格文件內容是否完整、一致和正確。

5-5-2 需求驗證技術-使用雛型

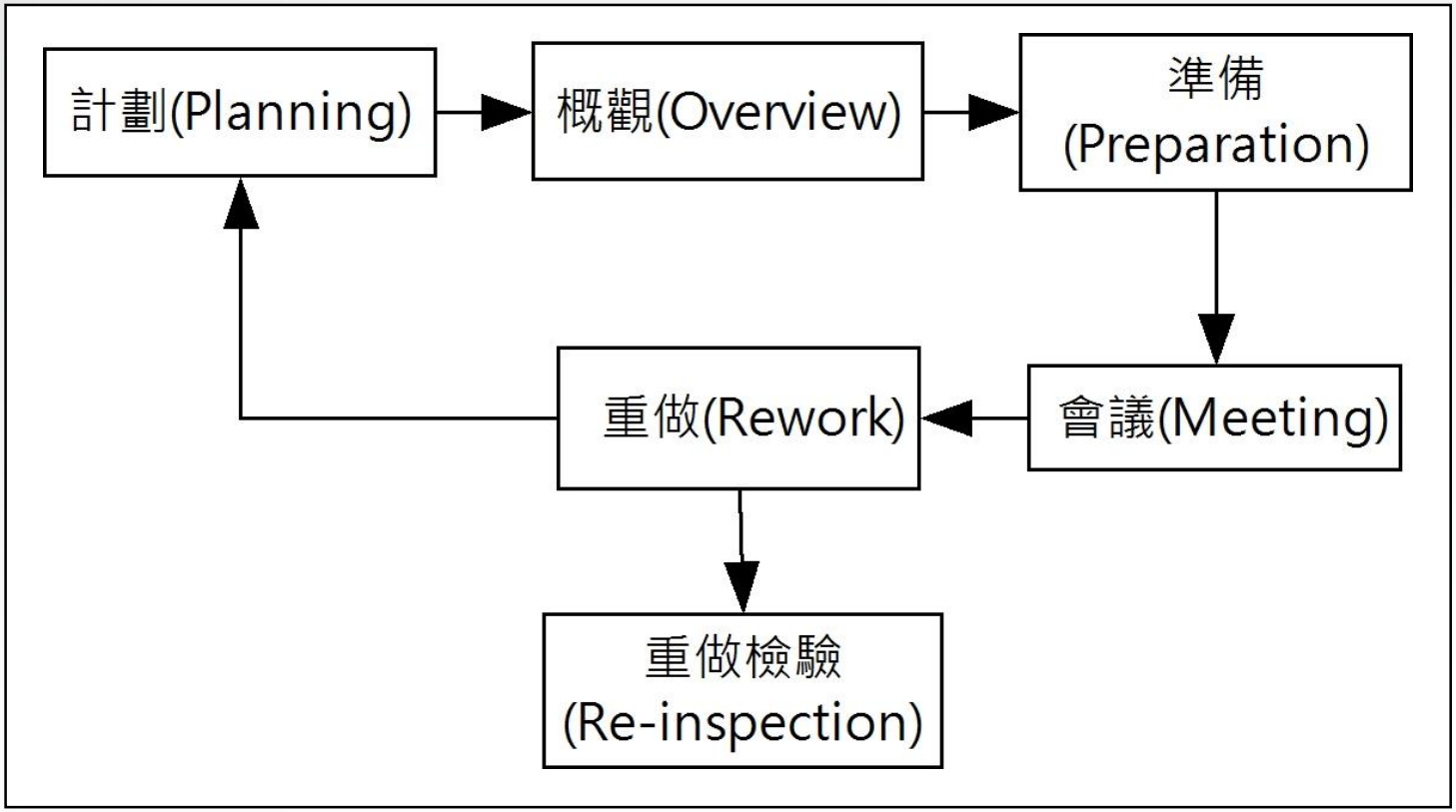
- 需求驗證可以使用雛型（Using a Prototype），也就是建立系統雛型來驗證功能性需求，或使用模擬器模擬系統功能來驗證功能性需求是否正確。

5-5-2 需求驗證技術-Fagan檢驗(說明)

- Fagan檢驗（Fagan Inspections）是Michael Fagan開發的軟體檢驗方法，它是使用系統化和結構化方式來找出文件錯誤，包含軟體開發每一個階段的開發文件，例如：程式碼、規格、設計或其他階段的文件。
- Fagan檢驗的目的是找出錯誤（Defect），它需要一個小組來執行，包含一位協調者的主席、一位記錄的秘書、欲檢驗項目的作者和一至多位檢驗員。

5-5-2 需求驗證技術-Fagan檢驗(階段)

- Fagan檢驗的流程包含六個階段，如下圖所示：



5-5-2 需求驗證技術-Fagan檢驗(階段說明)

- 計劃（Planning）：選擇檢驗小組的成員、準備文件和安排會議地點。
- 概觀（Overview）：針對欲檢驗項目替檢驗小組作一般性文件的簡報。
- 準備（Preparation）：每一位小組成員獨自研讀欲檢驗項目和支援文件，並且記下任何問題和可能錯誤。
- 會議（Meeting）：小組成員一起重新探討欲檢驗項目，以便真正發現錯誤。
- 重做（Rework）：由項目作者修正會議中發現且被小組成員認可的錯誤。
- 重做檢驗（Re-inspection）：協調者必須負責證實所有會議中發現的錯誤都已經更正。

End