

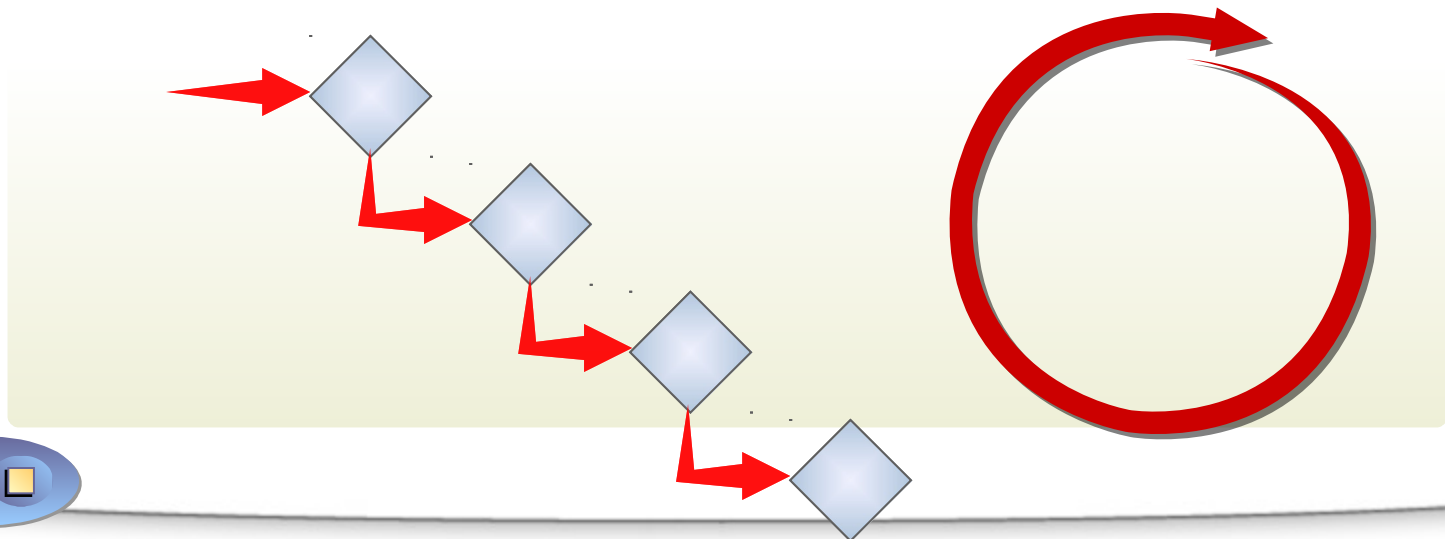
需求定義與解決方案規劃

錢達智 wolfgang.chien@gmail.com

Process Models

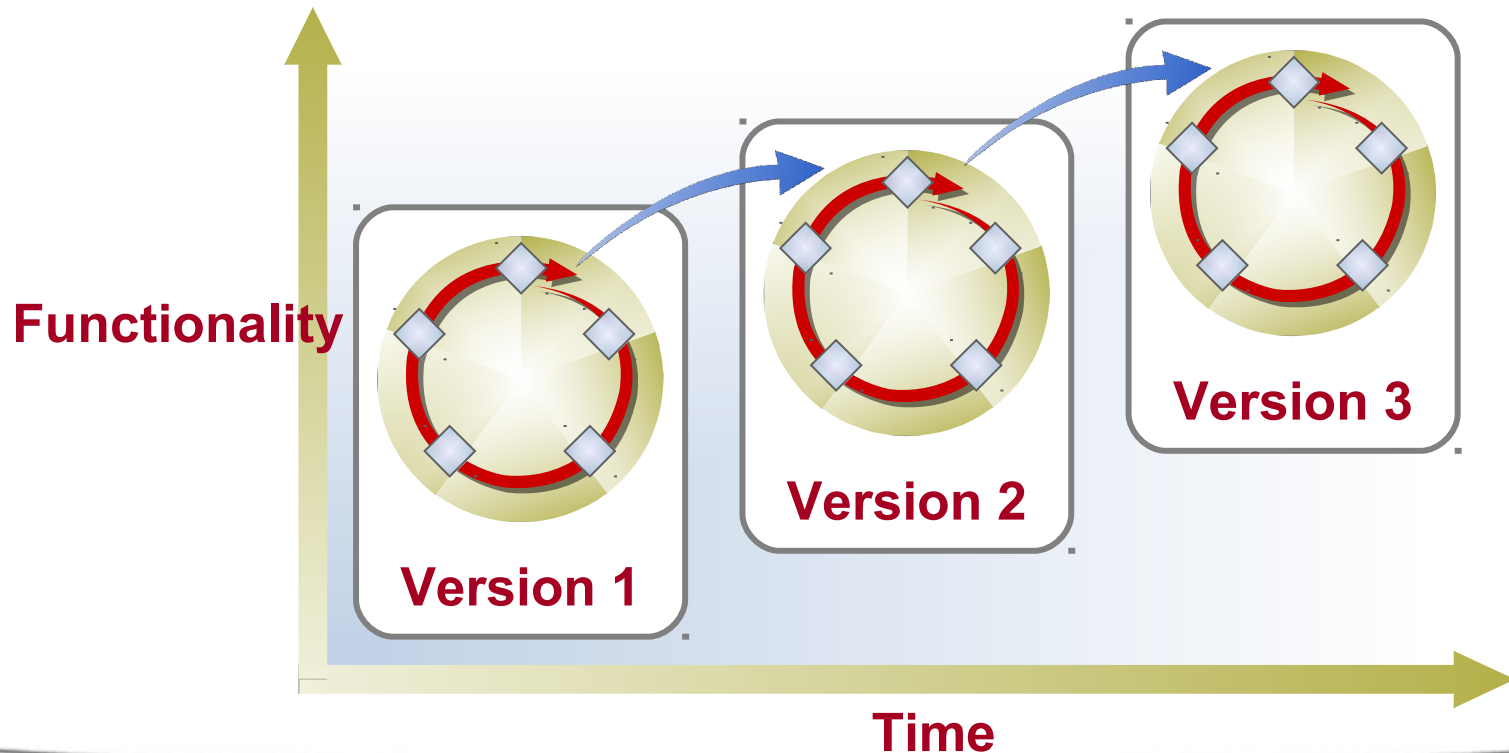
Process models 指的是軟體生命週期的活動指南。

- **Waterfall model:** 使用里程碑 (milestones) 代表一個階段告一段落，然後開始下一階段
- **Spiral model:** 不使用里程碑 (milestones)，採用漸近、快速、連續型的開發模式。

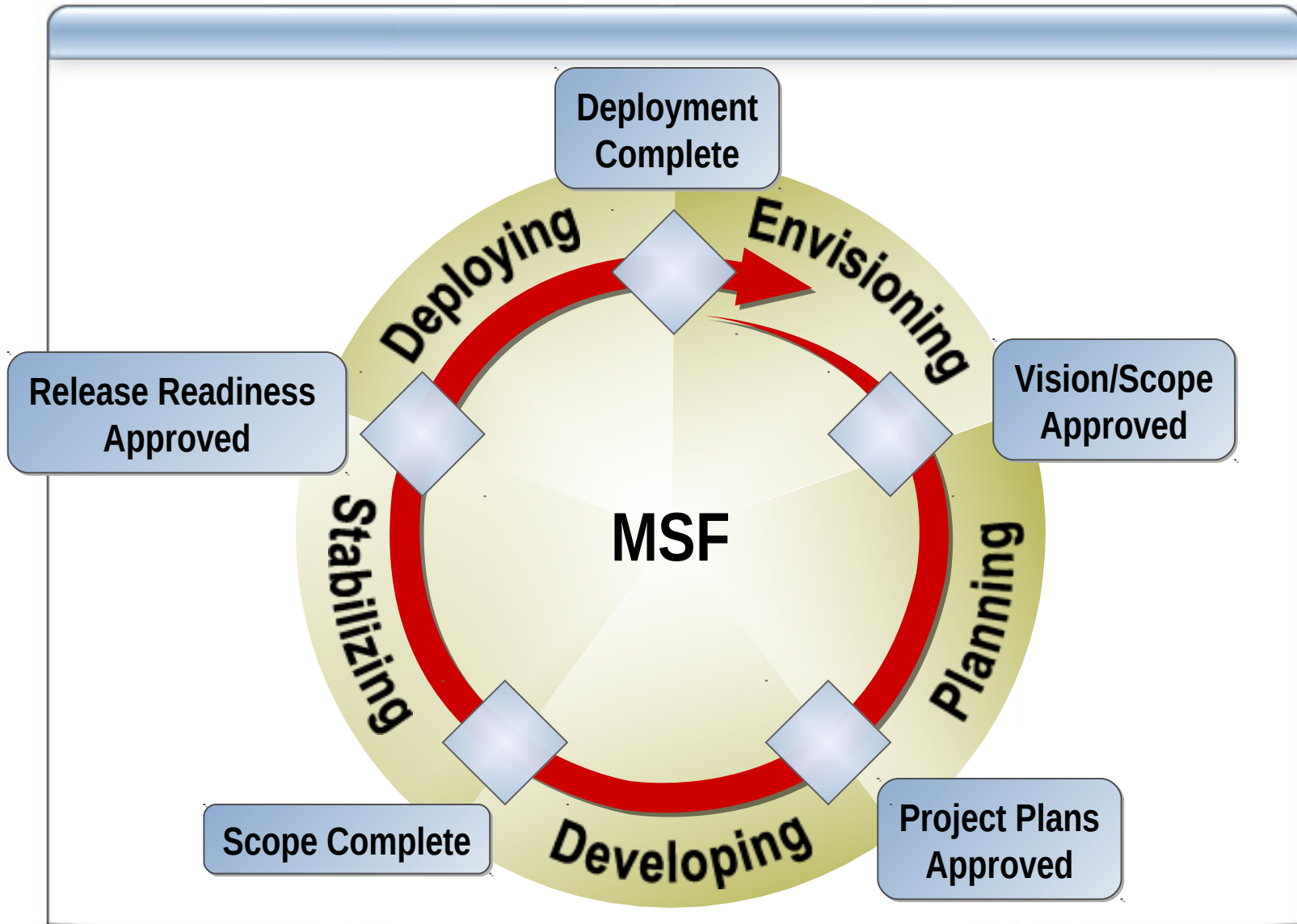


Iteration in Projects

- 將專案切成數個較小的目標
- 每個版本都是功能完整的可發行版本



MSF Process Model 的五個階段

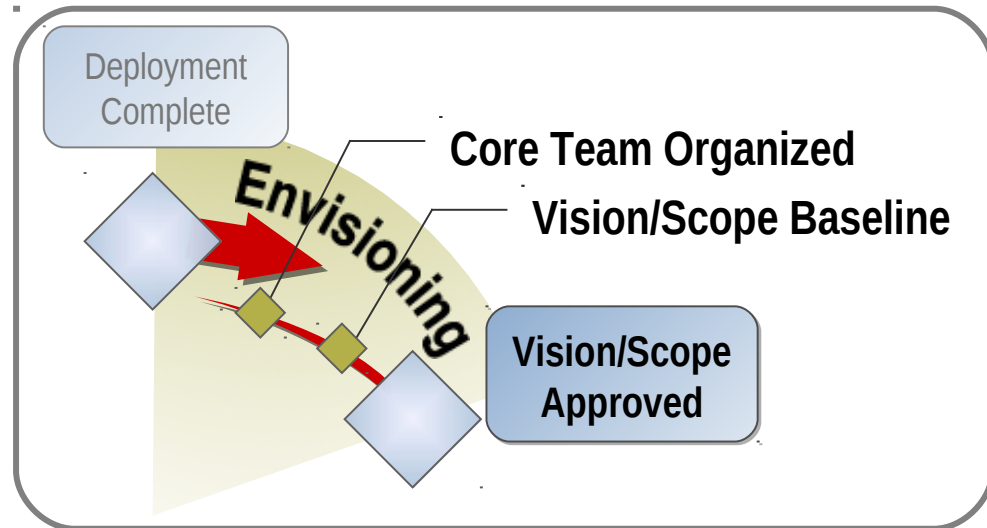


MSF 前兩個階段的工作及其 UML 圖形

- **Envision** (Use Case Diagram)
- **Conceptual Design** (Use Case / Sequence / Active / State Diagram)
(情節 / 故事)
 - 1-1 Research → 1-2 Analysis → 1-3 Optimization
- **Logical Design** (Class Diagram)
(模組 / 功能)
 - 2-1 Analysis → 2-2 Optimization
- **Physical Design** (Component / Deploy / Package Diagram)
(實作 / 細節)
 - 3-1 Research → 3-2 Analysis →
3-3 Rationalization → 3-4 Implementation

願景階段 (Envisioning Phrase)

- 組成核心成員
- 定義專案要解決的問題
- 設定專案的願景、範圍與目標



如何組織專案成員

- 知識
- 技術熟練程度
- 對方有空 / 願意來
- 預算
- 安全議題 (信得過的人)

關於問題 (Problem)

$$\text{Problem} = \text{Expectation} - \text{Current}$$

- 以「說出**想要什麼**」的方式來描述問題



錯誤訊息都不清不楚



- 錯誤訊息要有該如何處理問題的建議事項



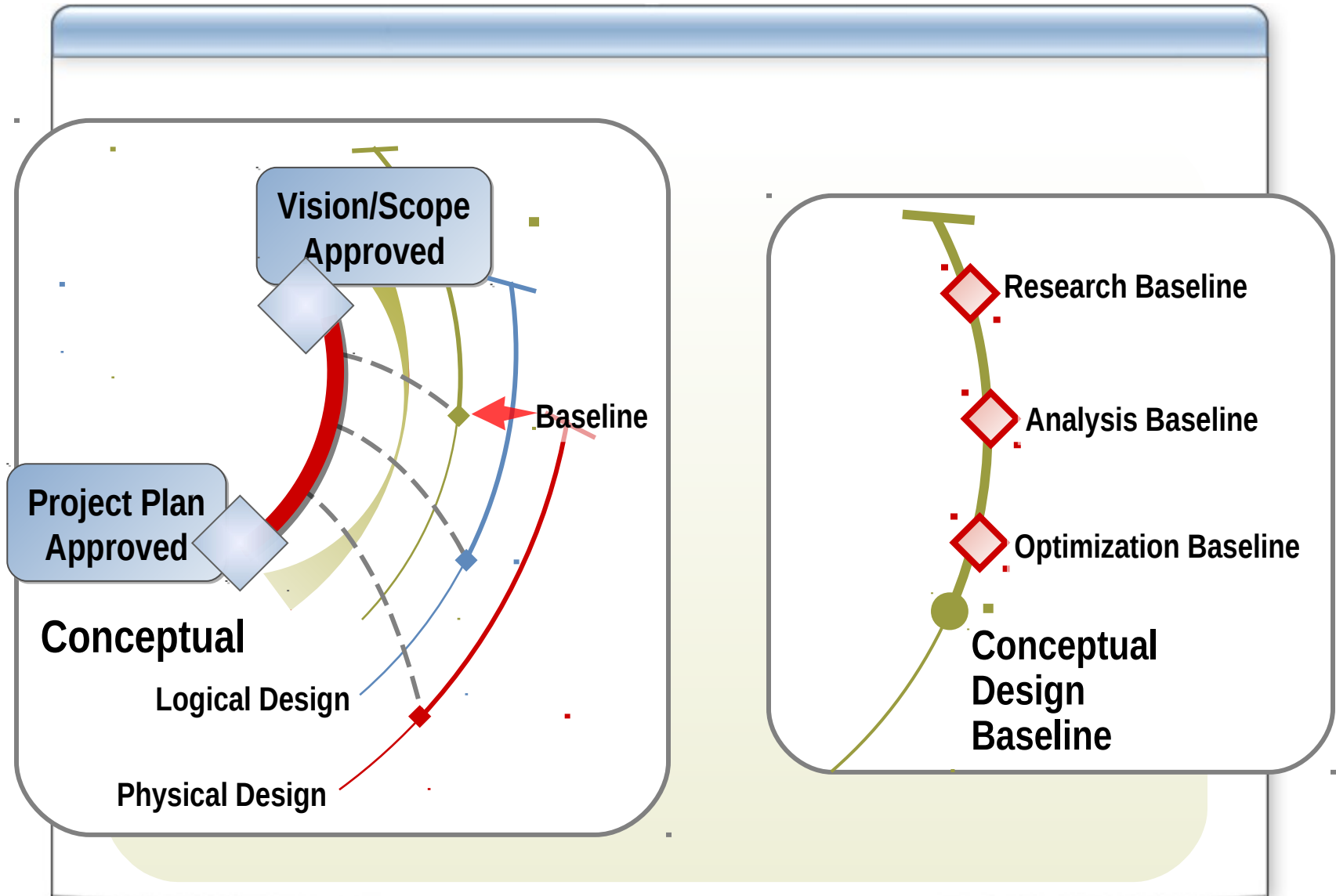
好慢喔！



- 系統應該要能處理每分鐘 250 人次的訂票服務



Conceptual Design (故事 / 情節)



Conceptual Design (情節) 的三個進程

- Research

- 調查什麼人，在什麼時間，做什麼事，怎麼做，以及為什麼這麼做

- Analysis

- 資料整理
- UML 圖形繪製
- 需求定義

- Optimization

- 發展未來情節：
什麼人，在什麼時間，做什麼事、怎麼做

資料蒐集

- 從三個來源：
 - 文件
 - 系統
 - 人員
- 採用次頁描述的各種方法
- 蒐集出下列資訊：
 - 商業
 - 應用程式
 - 操作流程
 - 技術

資料蒐集方法

| 方法 | 描述 |
|------|---------------------------------------|
| 觀察訪視 | 直接到現地觀察使用者如何工作 |
| 採訪 | 向特定使用者詢問資訊 |
| 群組討論 | 以座談會、網路論壇的型式討論與分享觀點 |
| 問券調查 | 設定、收集、統計意見 |
| 實地操作 | 在使用者的指導下，實際操作一遍 |
| 雛型系統 | 建構一個模擬系統試用看看（一張 A4 一個畫面，或者使用線框工具繪製草圖） |
| 錄音錄影 | 使用錄音錄影軟體 / 硬體記錄實況 |

定義需求

● 聆聽

- 問題 = 預期 – 現況。使用者正在「抱怨」、還是在講「怎麼做」，或者正在說「要什麼」。

問題是：他口頭上說的東西，真的是他要的嗎？

● 歸納

- 業務 A: 哪些是我們最好的客戶？
- 經理 B: 我需要地區別的銷貨統計
- 業務 C: 可以分析哪些產品賣得比較好嗎？
- 歸納：他們要銷貨分析統計，所謂客戶、產品、地區乃至時間，只是分析統計的四個面向而已。

● 問自己：「他(她)為什麼要這個？」

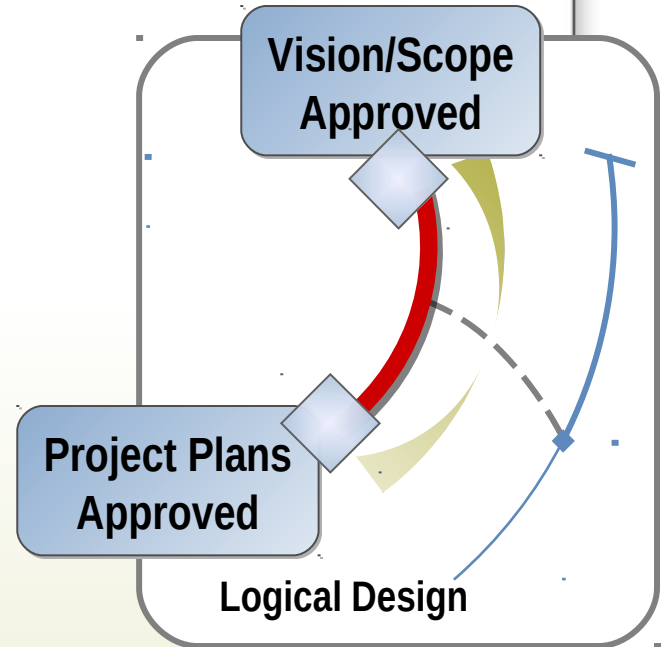
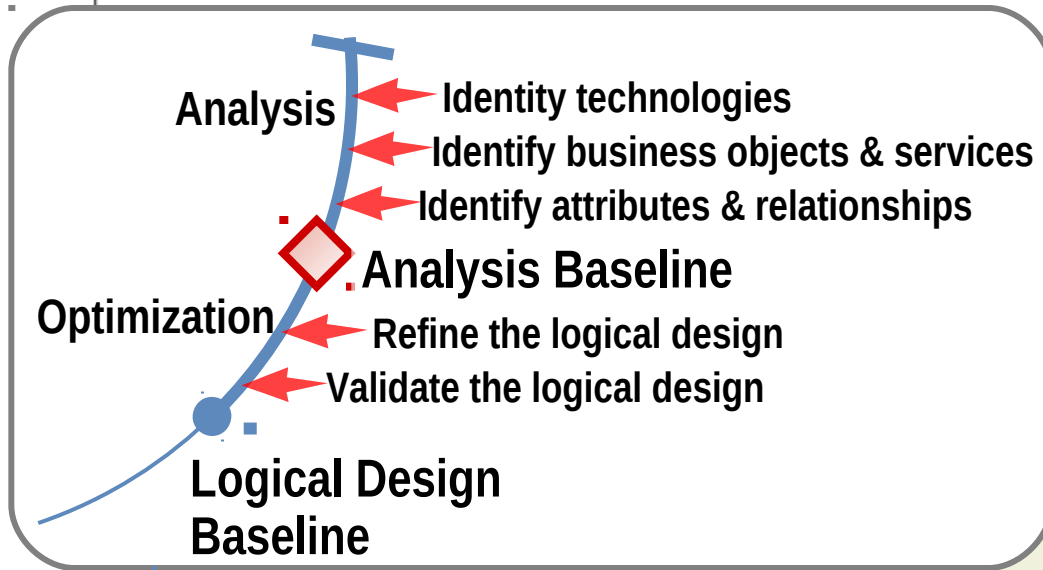
- 為什麼他要 A？那個理由比表面的 A 來得重要

● 時間

- 如果使用者又提一次甚至再三提到，表示使用者真的很在意

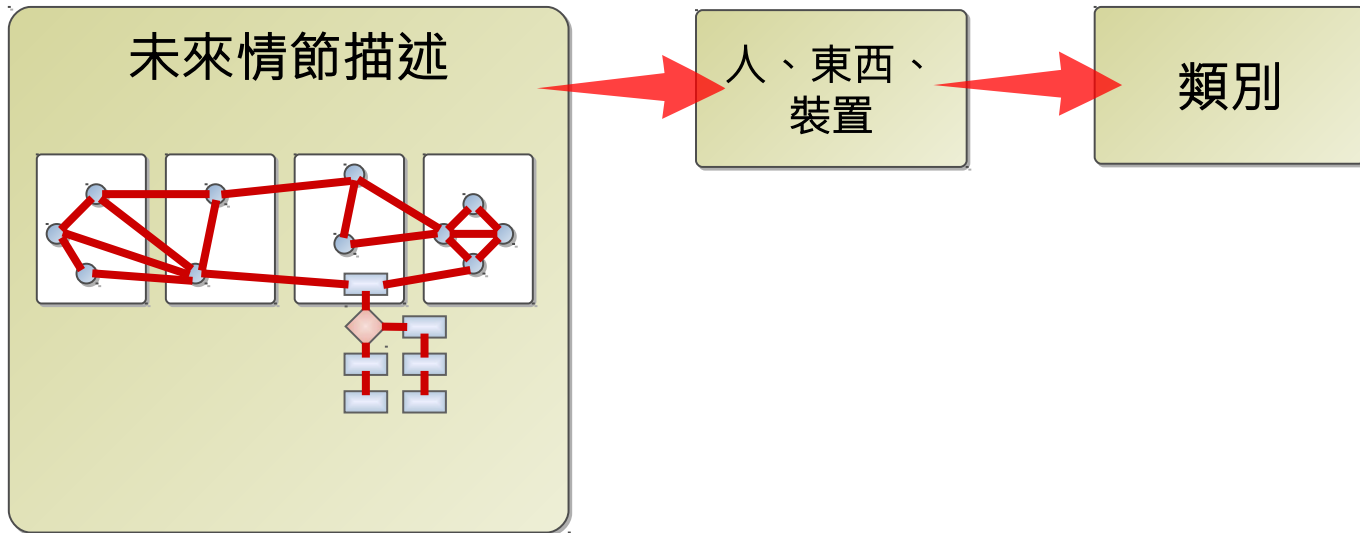
Logical Design (模組 / 功能)

- 以專案成員的觀點定義解決方案



如何定義類別 (Class)

檢視未來情節描述，留意「**名詞**」。這些名詞很可能就是類別名稱或屬性名稱。



如何定義屬性

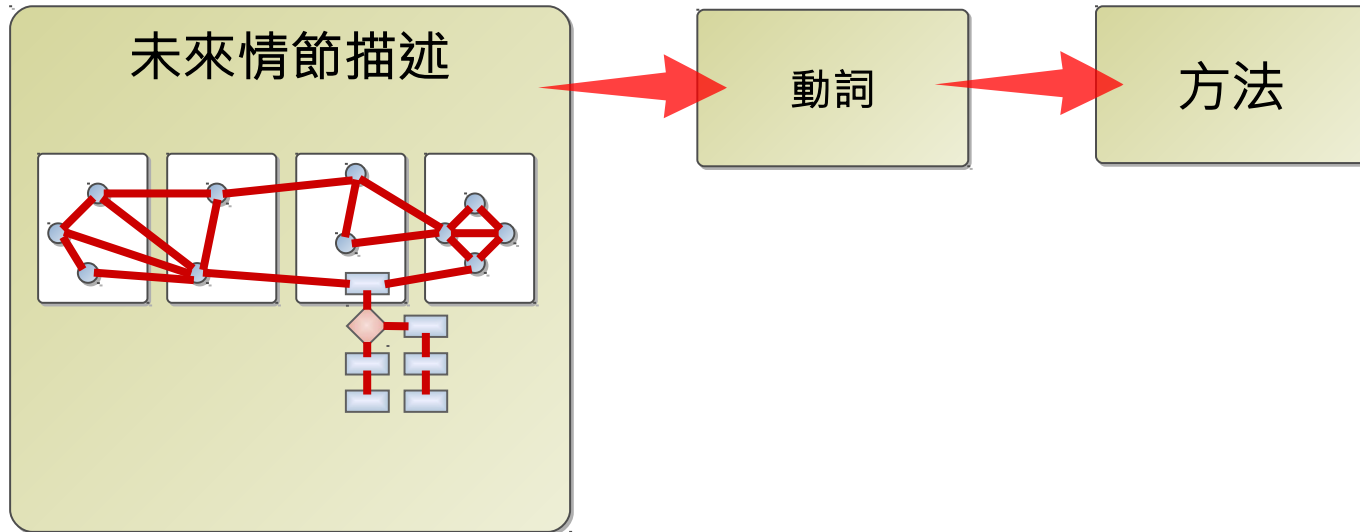
- 常識
- 互動性
- 主體性

Address 必須附屬於客戶類別， Address 是屬性

| 類別名稱 | 屬性 | 屬性值 |
|----------|-------------|----------------|
| Customer | CustomerID | 10076 |
| | Name | III |
| | Address | 台北市復興南路一段 |
| | ContactName | 劉小姐 |
| | Phone | (02) 6631-6666 |

如何定義方法 / 服務

檢視未來情節描述，留意「**動詞**」。這些動詞很可能就是類別的方法或系統需要提供的服務。



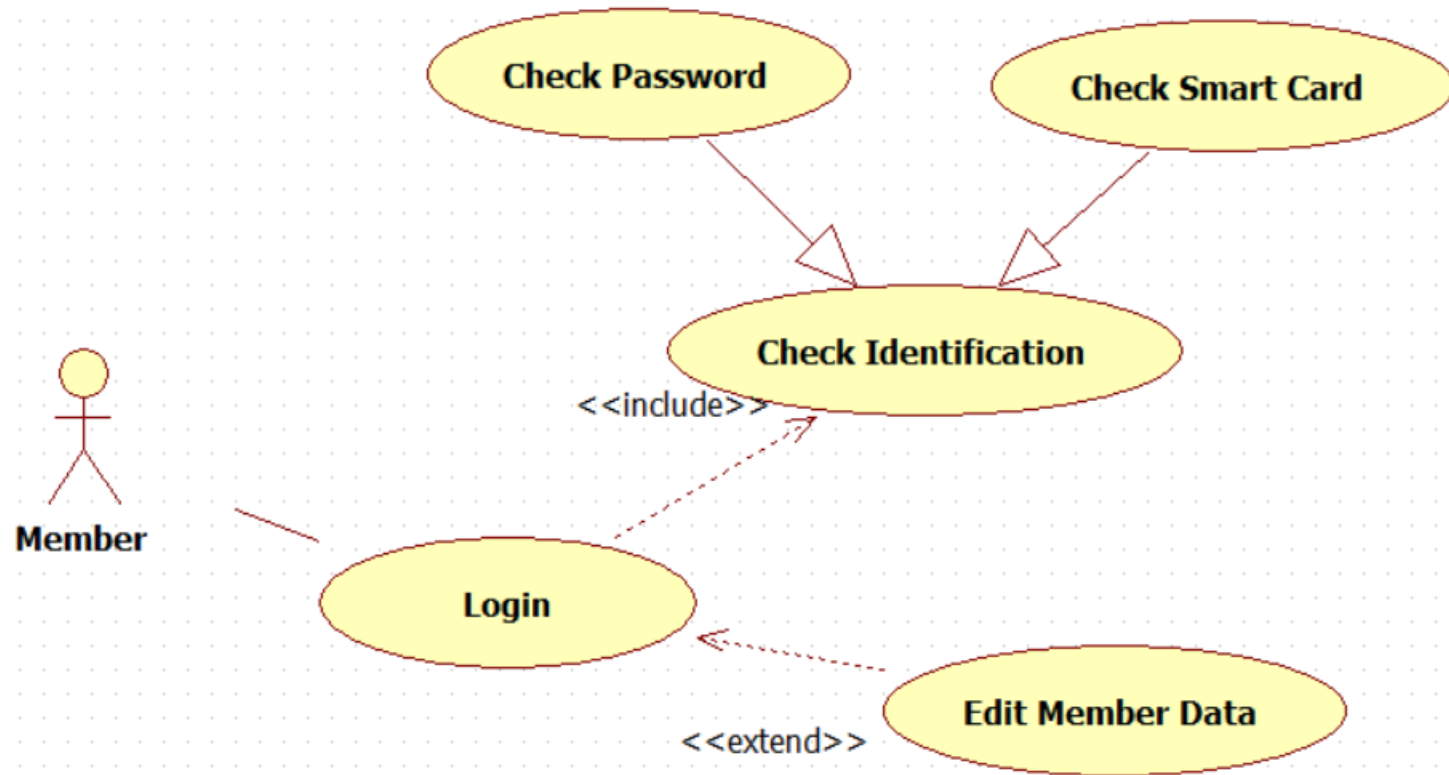
閱讀與討論

- 物件導向程式設計：為何說 composition 優於 inheritance ?
 - https://tw.twinkl.com/@arthurtw/*662v
- 要如何找出物件呢？（一）
 - <http://milikao.pixnet.net/blog/post/543590>
- 為什麼我找出來的物件都是 UI 物件？（二）
 - <http://milikao.pixnet.net/blog/post/543596>
- 到底誰該去繼承誰？（三）
 - <http://milikao.pixnet.net/blog/post/543592>
- 繼承是父子關係？才怪！（四）
 - <http://milikao.pixnet.net/blog/post/543716>
- 我不喜歡物件導向（來篇讓人反思的文章）
 - <http://blog.udn.com/yccsonar/7085309>

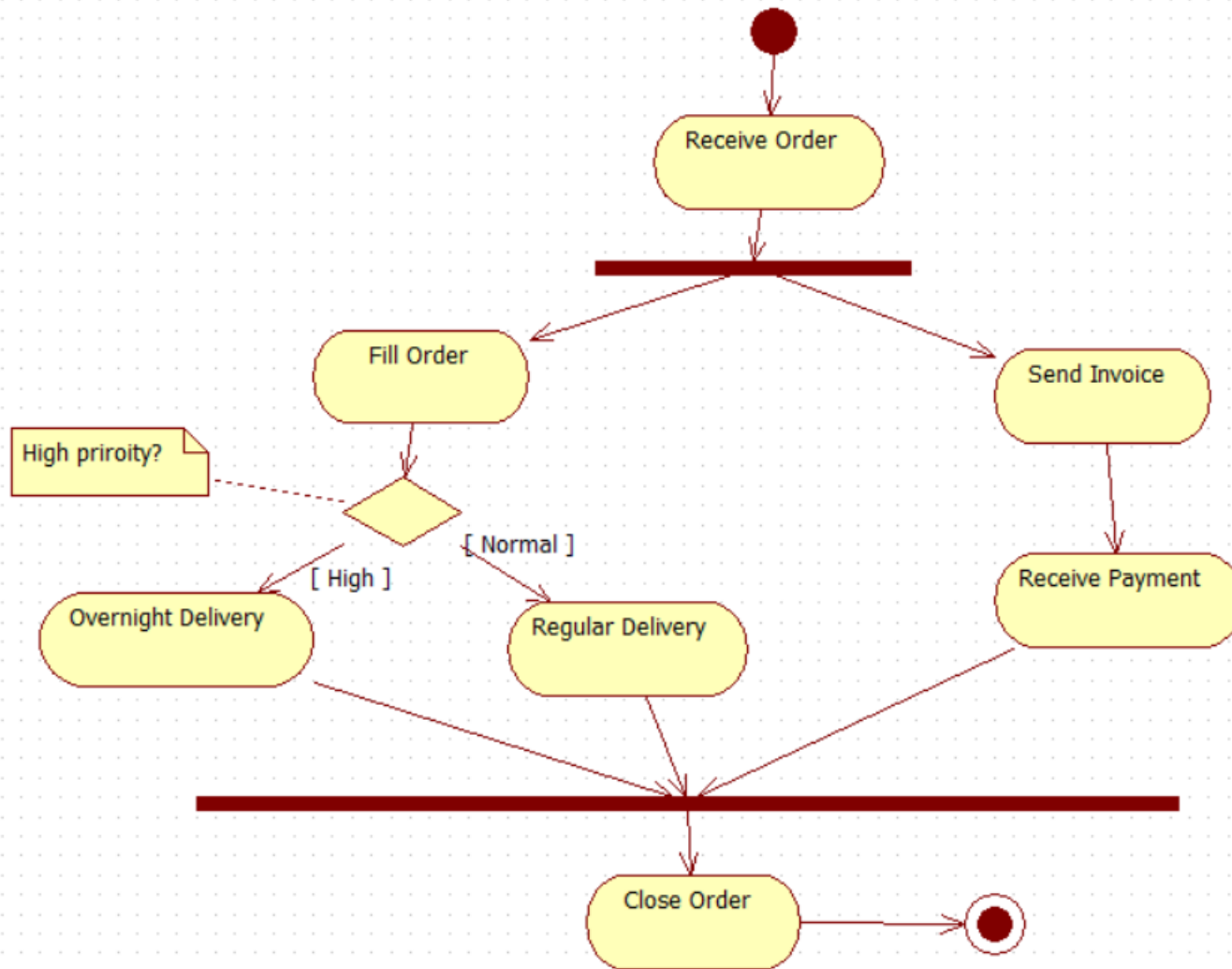
UML 圖形

- 每張圖形都有各自不同的觀點：
 - Class
 - Object
 - Use case
 - Component
 - Deployment
 - Collaboration
 - Sequence
 - State

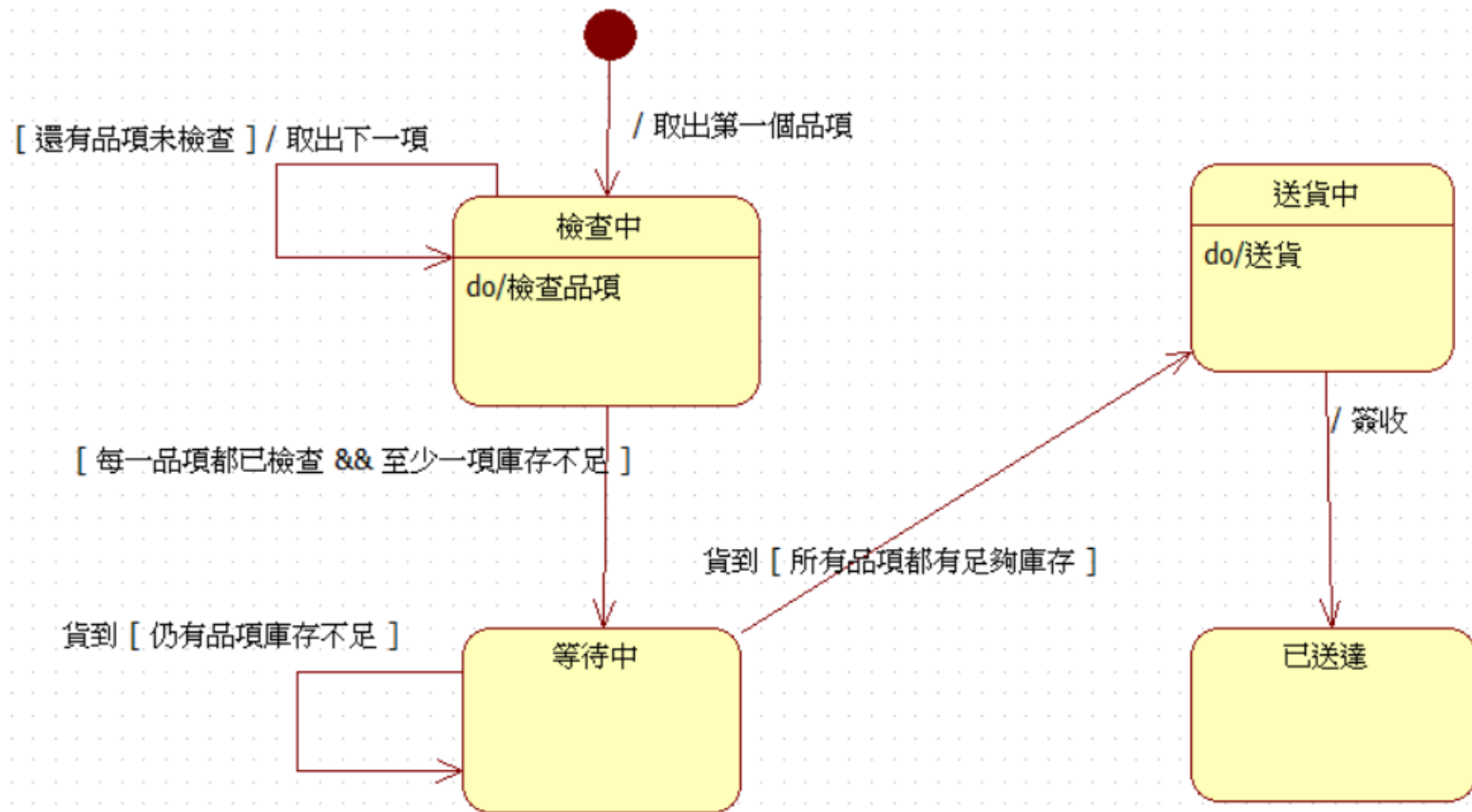
UML 圖例 : Use Case



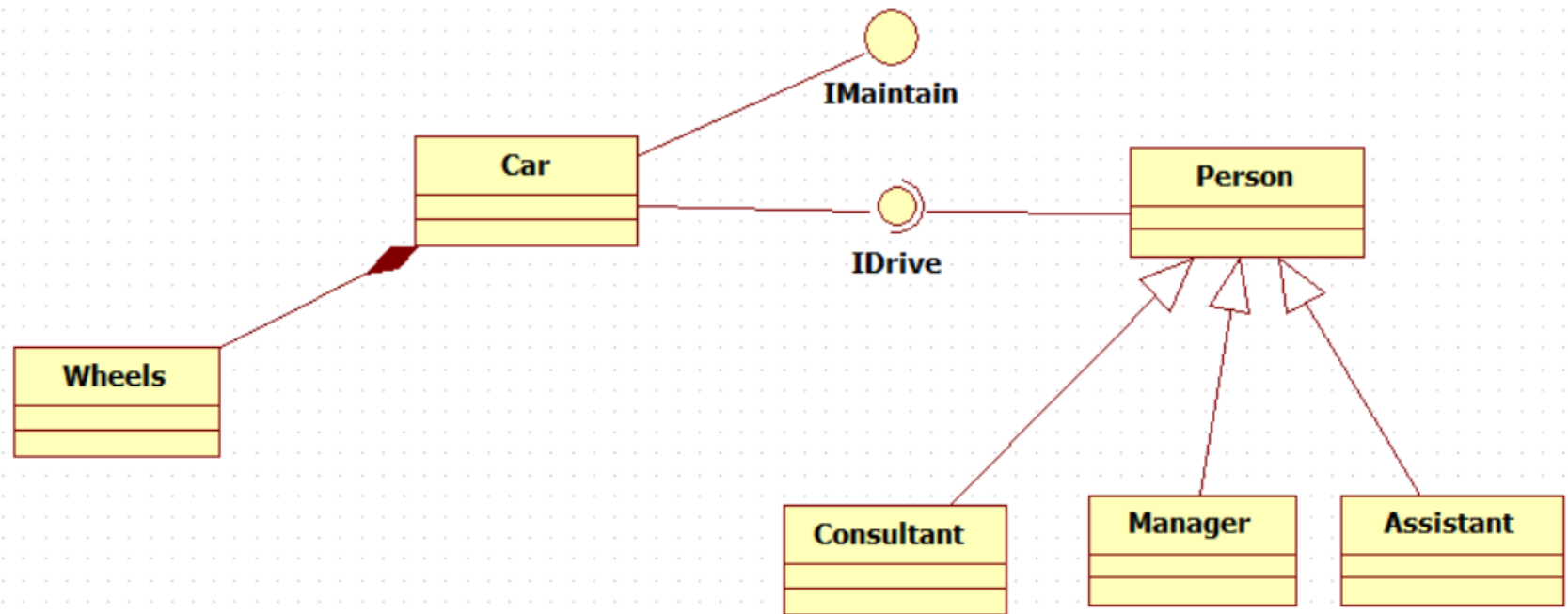
UML 圖例 : Active Diagram



UML 圖例：State Diagram



UML 圖例 : Class Diagram



UML 圖例 : Sequence Diagram

