

搞笑談軟工


敏捷開發，設計模式，精實開發，Scrum，軟體設計，軟體架構

2014年4月28日 星期一

SOLID：五則皆變

SOLID物件導向設計原則

- **S**ingle responsibility
- **O**pen-closed
- **L**iskov substitution (Subcontracting)
- **I**nterface segregation
 - many client-specific interfaces are better than one general-purpose interface.
 - Clients should not be forced to depend on methods that they do not use.
- **D**ependency inversion



Copyright ©2012-2014 Teddysoft

今年4月19日在上「Design Patterns 這樣學就會了進階實作班」Day 1 課程的時候，Teddy 介紹了 SOLID 這五個物件導向設計原則。這五個原則出自於《Agile Software Development》這本書，就 Teddy 所知，Open-Closed Principle 與 Liskov Substitution Principle（又稱為 Subcontracting）這兩個原則並非作者原創，因為 Teddy 曾經在《Object-Oriented Software Construction, 2nd》這本書讀過這兩個原則，至於其他三個原則就沒有在比這本書作者還早提出的其他文獻中讀過。

這五個原則的其中四個，Teddy 曾經在部落格介紹過：

- 〈亂談軟體設計（2）：Open-Closed Principle〉
- 〈亂談軟體設計（3）：Single-Responsibility Principle〉
- 〈亂談軟體設計（4）：Liskov Substitution Principle〉
- 〈亂談軟體設計（5）：Dependency-Inversion Principle〉

今天想要談一個問題，這五個原則都在講同一件事，請問是什麼事？

鄉民甲：明明就是五個不同的原則，怎麼會講同一件事？

原本 Teddy 也沒意會到這件事，就在上課前一天的晚上，Teddy 在家裡備課的時候，把《Agile Software Development》這本書針對這五個原則的說明又讀了一次，才發現到，

原來這五個原則，都在談「**改變**」這件事。

講的清楚一些，它們在談「**面對原始碼改變的五種不同策略**」，或是說「**從五種不同的角度，來應付、管理原始碼改變**」。

哪五個角度？

- **SRP**：降低單一類別被「改變」所影響的機會，在《Refactoring: Improving the Design of Existing Code》這本書中所提到的 Divergent Change（發散式變化）就是一種類別違反 SRP 所形成的壞味道，類別設計符合 SRP 便可避免 Divergent Change。
- **OCP**：當新增需求的時候（功能變化），在不改變現有程式碼的前提下，藉由增加新的程式碼來實作新的功能。
- **LSP**：LSP 談的是繼承，也就是透過繼承時子類別所造成的「行為變化」要如何設計，才可以讓程式在 runtime 透過多型所繫結（bind）到的子類別實作，不至於違反父類別介面的合約。
- **ISP**：針對不同的客戶端，只開放其所需要的介面讓它使用，如此一來可避免與降低客戶端因為不相關介面改變也一起被迫需要改變的問題。假設你有一個擁有 10 個函數的類別，現在有三個不同的客戶端都需要使用這個類別其中的 4、7、5 個函數。如果直接把這個類別傳給三個不同的客戶端，則這些客戶端很可能會因為它所沒使用到的函數改變了，也被迫跟著改變（因為原本類別的介面改變了）。另一種作法，則是針對三個不同的客戶端，提供三個不同的 Adapter。透過 Adapter，只開放客戶端所需的函數給它們，以隔離因為不相關介面改變所造成的客戶端改變。
- **DIP**：DIP 談的是 caller 和 callee 之間的關係，在兩者之間增加一個（抽象）介面，避免 caller（上層程式）因為 callee（底層程式）改變而被迫改變。

個別來看，**SOLID Principle** 是五個不同的設計原則。從「對付改變」的角度來看，它們就可以被歸為同一類的設計原則，只不過因為造成改變的原因不同，所以有五種不同的應對策略。

友藏內心獨白：這麼簡單的道理，居然最近才參透啊🤔。

張貼者：Teddy Chen 於 上午 12:00