

Week 8-專題結果分享



程式加密及版本管理介紹

/ 如何發展大程式碼？

個人經驗，對自動化程式而言，**100**行以內算是小程式。中型程式則是**100-1000**行，超過**1000**行可以算是大程式。這裡的行數不包含空白行，註解。開發小程式不太需要考慮結構的問題。中型程式就必須以函數，甚至是物件導向來組織程式碼。大型程式會需要模組化。程式規模越大，除錯困難度也越高，因此需要考慮單元測試。

- 使用者溝通
- 模組化
- 單元測試
- 版本管理
- 保護及加密

溝通的重要性

如果程式使用者跟程式開發者是同一個人，由於不需要溝通成本，開發速度是最有效率的。多數的時候，使用者與開發者為不同人，此時溝通的成本無可避免。很多開發者疏忽了溝通的重要性，等到程式開發完成才發現程式不符使用者期望，導致整個專案最後以失敗告終。

這種問題其實是可以避免的，最好的方法是開發過程中持續的與使用者討論，也可以透過原型展示來輔助討論的進行。也就是先完成操作介面的設計，如果符合客戶的期望，再進行核心代碼的開發。



／ 溝通的困難

相信大家都有經驗在跟家人解釋工作的內容時，極其困難。但是跟大學同系所的同學解釋時就簡單的多，跟其他部門的同事只要幾句話就可以說明清楚。這個差異來自知識重疊的多寡。知識重疊越多，溝通成本越小。

因此在評估自動化程式開發的時候，除了必須考慮技術的掌握度，更重要的是跟使用者的契合度，幫自己同部門的同事開發軟體的困難度通常要比幫其他部門開發小的多，除了技術問題，彼此之間的默契與信賴讓溝通可以有更有效的進行也是很重要的原因。



解析Python模組(Module)和套件(Package)的概念

- 當我們在開發大型應用程式時，如果沒有適當的組織程式碼，除了會降低開發的效率外，也不易於維護，所以模組(Module)化就顯得相當的重要，讓程式碼能夠透過引用的方式來重複使用，提升重用性(Reusable)。
- 但是隨著專案模組(Module)的增加，將難以管理及問題的追蹤，這時候就能將模組(Module)打包成套件(Package)，利用其階層式的結構來彈性規劃模組(Module)。

單元測試(Unit Testing)[WIKI]

- 在電腦編程中，單元測試又稱為模組測試，是針對程式模組（軟體設計的最小單位）來進行正確性檢驗的測試工作。程式單元是應用的最小可測試部件。在程序化編程中，一個單元就是單個程式、函式、過程等；對於物件導向程式設計，最小單元就是方法，包括基礎類別（超類）、抽象類、或者衍生類別（子類）中的方法。
- 通常來說，程式設計師每修改一次程式就會進行最少一次單元測試，在編寫程式的過程中前後很可能要進行多次單元測試，以證實程式達到軟體規格書要求的工作目標，沒有程式錯誤；雖然單元測試不是必須的，但也不壞，這牽涉到專案管理的政策決定。

軟體版本控制

軟體設計師常會利用版本控制來追蹤、維護原始碼、檔案以及設定檔等等的改動，並且提供控制這些改動控制權的程式。在最簡單的情況下，軟體設計師可以自己保留一個程式的許多不同版本，並且為它們做適當的編號。這種簡單的方法已被用在很多大型的軟體專案中。該方法雖然可行，但不夠有效率。除了必須同時維護很多幾乎一樣的原始碼備份外；而且極度依賴軟體設計師的自我修養與開發紀律，但這卻常是導致錯誤發生的原因。 [WIKI]

最受歡迎的版本控制技術與網站為git及GitHub，參考：

[Git與GitHub介紹，軟體版本控制基本教學](#)



Python

- Object Oriented Programming
- Module and Packate
- Numpy & Scipy
- Unit Test
- Machine Learning
- ...

AEDT

- ODB
- Batch Modeling and Post Processing
- ACT
- UDP (User Defined Primitive)
- UDO (User Defined Output)
- ...

如何不讓別人看到程式碼？

有兩種手段:

- 編譯成.dll
- 包裝成ACT(ANSYS Customization Toolkit)

成果以及開發經驗分享

分享原則

- 每人以5分鐘上限
- 內容包括
 - 自動化程式展示
 - 程式效果說明
 - 學習及開發心得
 - 未來發展方向



 **Ansys**

