**Software Configuration Management(SCM)介紹**

**前言**

在軟體開發時會經常有變更的動作，但此變更是大家都能上去做更改的，因此時常會造成開發人員之間的混亂或不知道修改到哪去…除了這個，最大的麻煩是工作成果可能會被覆蓋或無法回溯，這對開發人員來說真的是一大困擾阿…

SCM的誕生就是用來造福各大開發人員及公司的呀!

**定義**

SCM(Software Configuration Management，軟體配置管理)是一種通過執行版本控制及控制修改的系統，最原始的方法是採用手動管理版本的方式，而版本控制是拿來記錄文件內容變化，有利於開發者未來要查閱版本修訂情況的系統。

**問題**

假設當一個新版本產生時，給它命名為A，之後要修改時把A先複製起來，在複製的資料夾內進行修改，這樣A版本就被保存下來也不會遺失了。雖然這種方式看似簡單且可以解決回溯及覆蓋問題，但其實問題很多!

1. 如果保留(複製)太頻繁，會造成有大量且重複內容的資料夾存在，除了管理麻煩外，時間一長，可能會造成檔案毀損，版本無法回溯的問題。
2. 因為保留太多資料夾，容易造成版本混亂，開發人員用錯版本都是有可能的!

**解決方法**

為了解決上述的問題，在很久以前程式設計師就開發了很多版本控制系統，大多採用簡單的資料庫來記錄文件內容變化，但簡單的資料庫卻無法滿足開發人員共同編輯的問題，因此就有了集中化的版本控制，由一個伺服器來管理所有版本的檔案，開發者只要連到這台伺服器取出檔案就能做使用，也不會有版本混亂的問題。

有集中化的本版控制系統，怎能少了分散式版本控制系統呢!!

分散式系統顧名思義就是指當中央伺服器掛掉時，那段時間會所有的開發人員都無法動作，此時若沒有備份，更慘的可能連資料都會遺失，因此分散式系統就是來解決這個問題的!

**步驟**

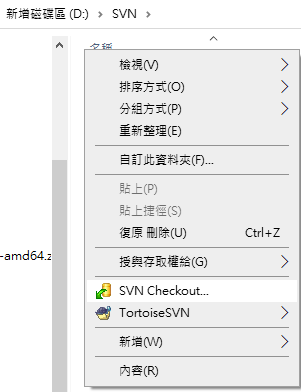
SCM的三大步驟為

1. Get Latest Version
2. Check Out

假設我要check out這個版本的專案，把底下藍色網址複製起來



且在相對應的位置check out。



3.修改完後Check In

(看起來多了三個步驟，但其實不包含修改，加起來也才幾秒鐘的事情，比一個一個複製來的快多了)