



版本控制系統(Version Control System)

Part 1 簡介

Wei-Tsung Su
suwt@au.edu.tw
564D





歷史版本

版本	說明	日期	負責人
v1.0	中文化初版	2019/02/26	蘇維宗





參考文件

1. http://en.wikipedia.org/wiki/Revision_control
2. <http://homes.cs.washington.edu/~mernst/advice/version-control.html>
3. <http://git-scm.com/book>
4. <https://backlog.com/git-tutorial/tw/>





內容

- 版本控制
- 版本控制系統(Version Control System, VCS)
 - VCS的類型
 - VCS專有名詞
 - VCS面臨的挑戰
- 免費的VCS服務
 - 網頁VCS服務
 - 免費工具





版本控制

版本控制是對於文件、電腦程式、大型網站、與其它資訊之變更的管理。

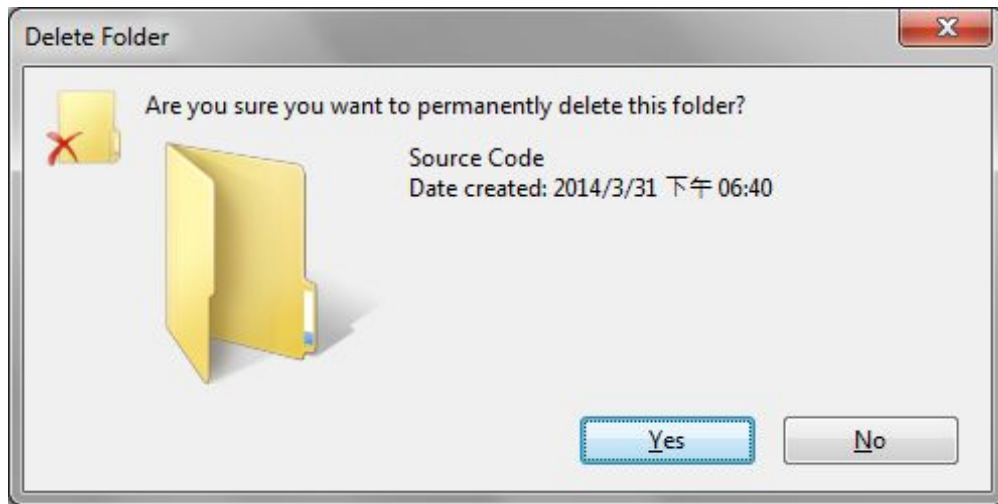
Source: Wikipedia

在版本控制中，通常每次變更都會以一個"版本號碼(revision number)"來辨識(例如，最新的版本是1.3版)。



為什麼要使用版本控制?

有時候, 你可能手殘砍掉整個專案...



為什麼要使用版本控制? (續)

雞同鴨講的狀況常常發生 ...

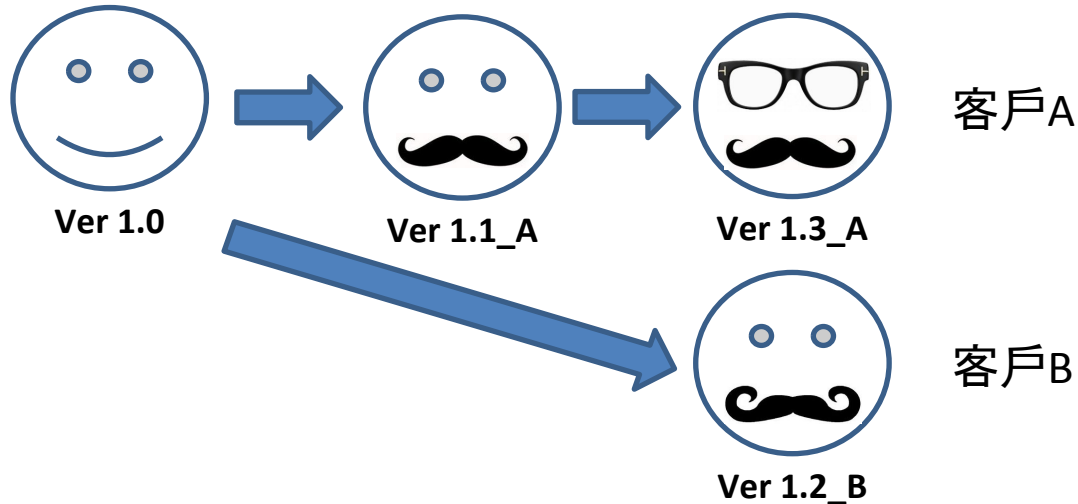
版本1.1
`void LOOP() {
 for(;;);
}`
移除for迴圈!

版本1.0
`void LOOP() {
 while(true);
}`
哪來的for迴圈?



為什麼要使用版本控制? (續)

許多客戶都想要客製化功能 ...





如何避免這些問題?

版本控制有以下幾個特性

- 可以記錄檔案的所有版本
- 提供多人協同開發的環境
- 可以提供產品的不同版本
 - 不同平台(例如, Windows、Mac、Linux等)
 - 不同客戶
- ...





版本控制系統(Version Control System)

版本控制系統(VCS)是一種軟體系統可以用來記錄檔案(或一組檔案)隨著時間改變的各個版本以利後續可以取得特定版本。

Source: *Pro Git*

VCS的種類

本地式VCS (例如, [rcs](#))

集中式VCS (例如, [CVS](#)與[SVN](#))

分散式VCS (例如, [Git](#))



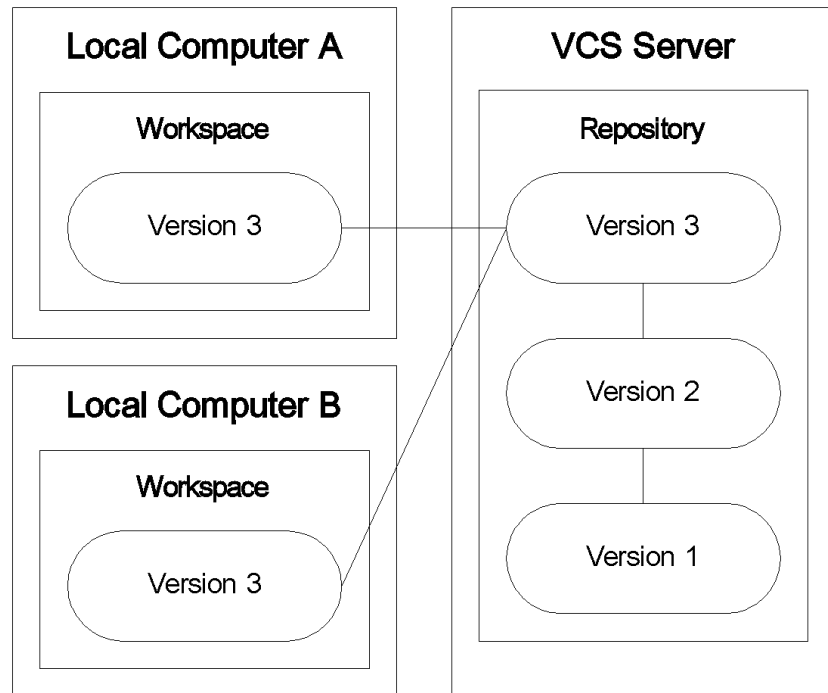
VCS專有名詞

Workspace

開發者開發中的專案版本
(儲存在本機)

Repository (簡稱Repo)

包含專案的所有版本
(可儲存在本機或遠端)



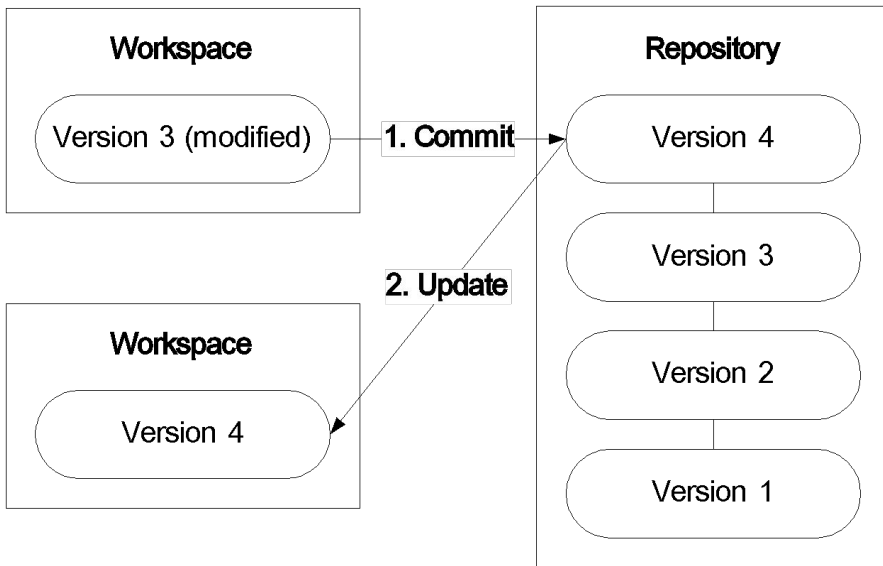
VCS專有名詞(續)

Commit

將Workspace的變更同步到Repo

Update

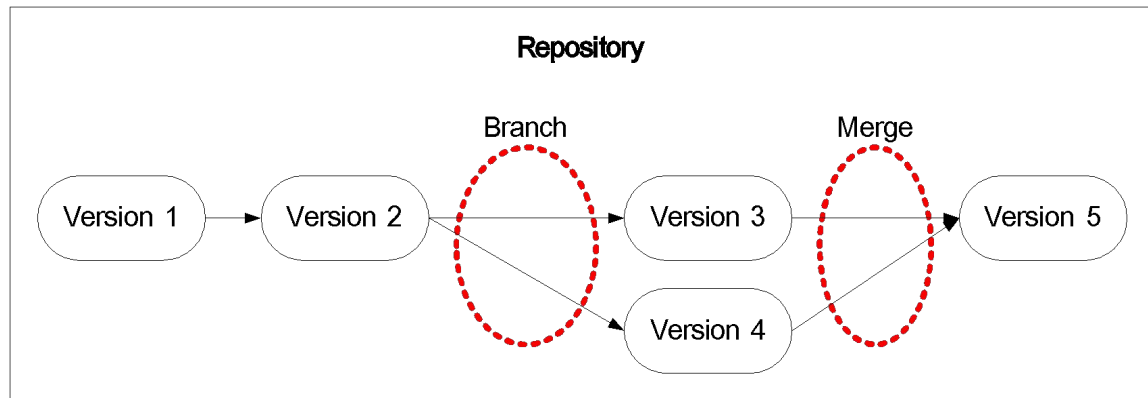
將Repo更新到Workspace



VCS專有名詞(續)

Branch (分支)

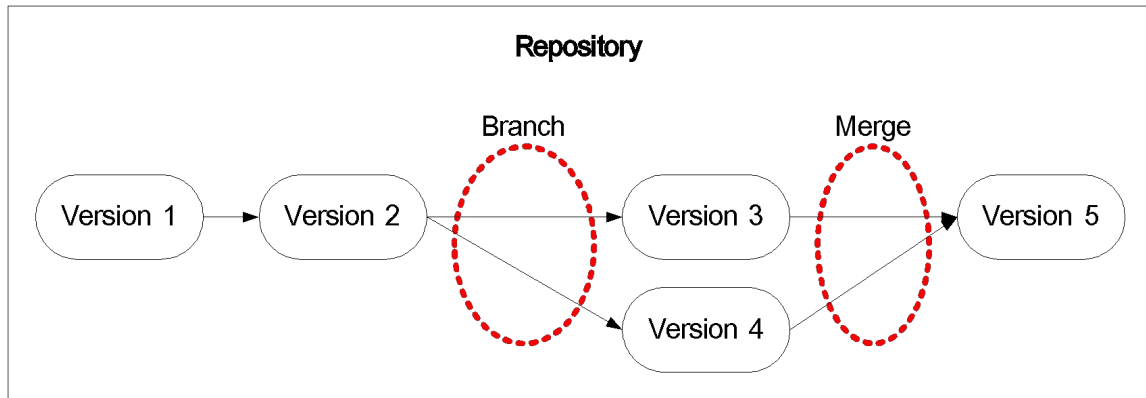
專案的版本可能會因為特定原因會產生分支並以不同的速度與方式進行更新。



VCS專有名詞(續)

Merge (合併)

將兩個版本分支合併成一個新的版本。



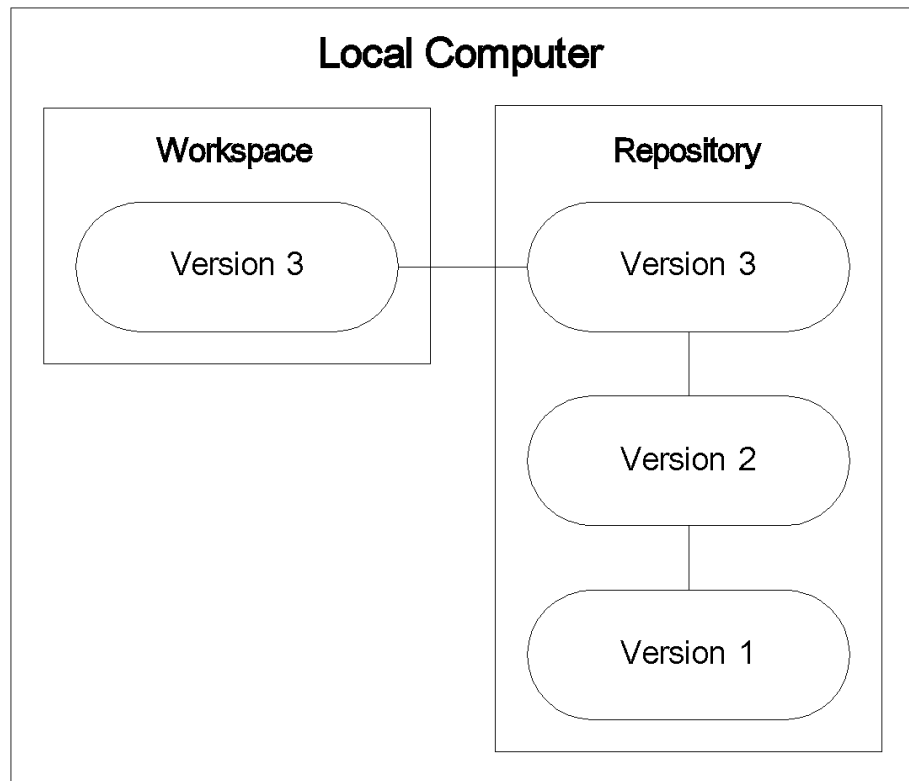
本地式VCS

Repo只儲存在本地端。

適合單人開發的專案。

例如,

Revision Control System ([RCS](#))



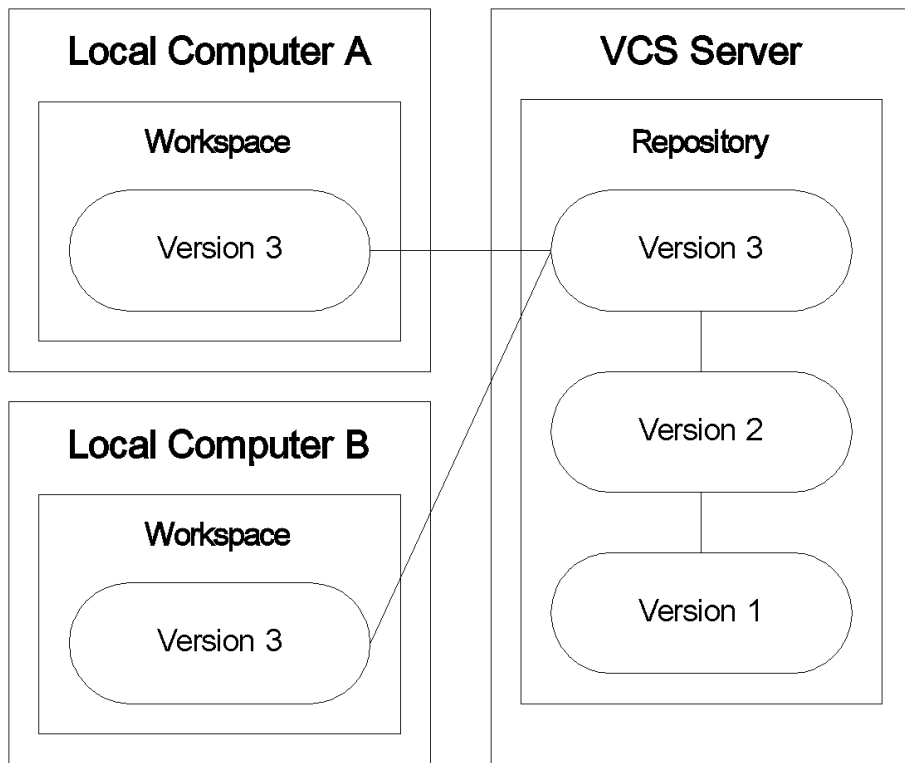
集中式VCS

Repo只儲存在遠端。

適合多人開發，缺點是切換/更新版本時需要連網。

例如，

Apache Subversion ([SVN](#))

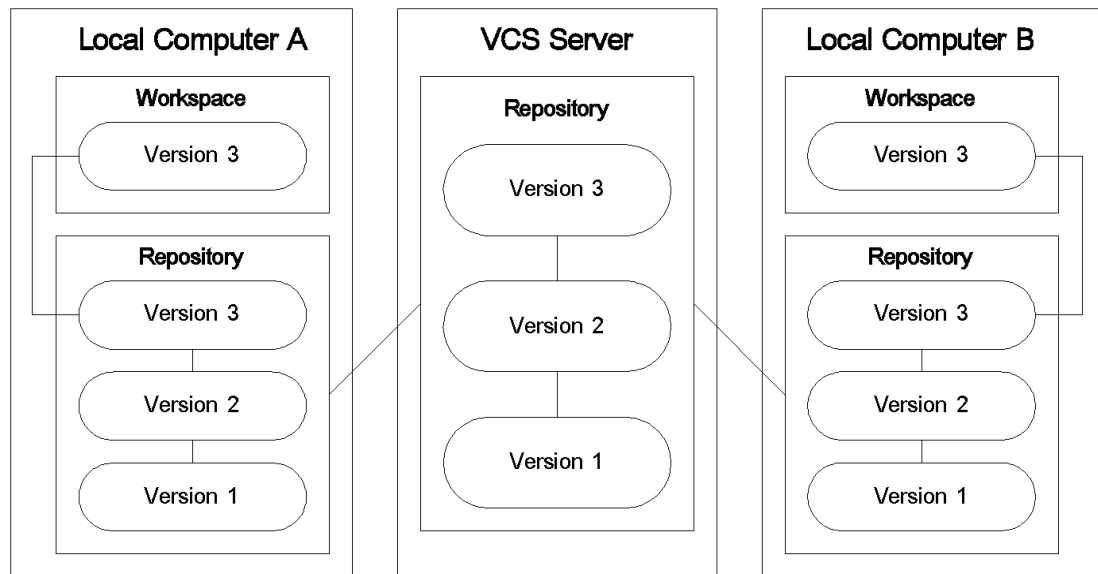


分散式VCS

Repo儲存在本機與遠端。

適合多人開發，切換/更新版本時不須連網但儲存空間較大。

例如，[Git](#)

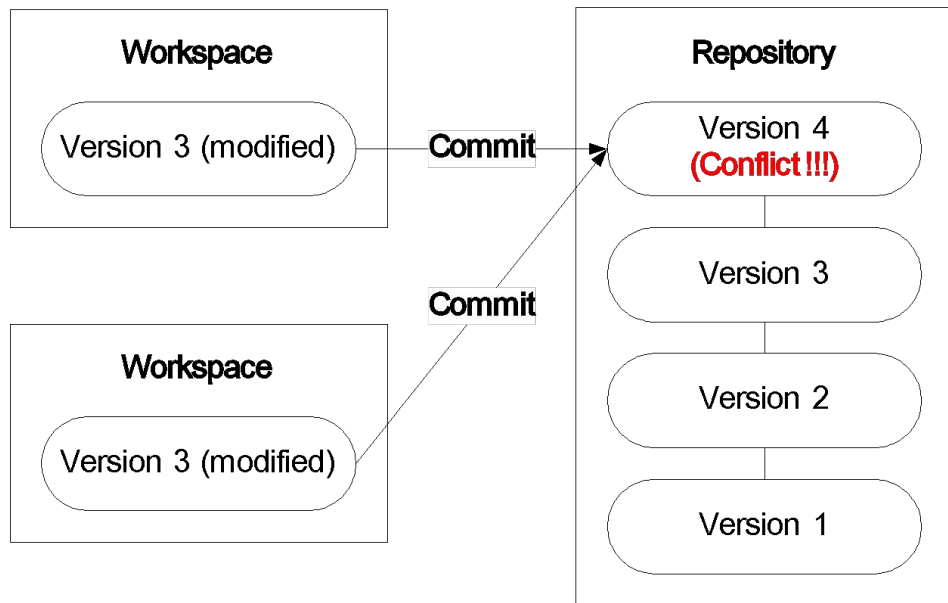


VCS面臨的挑戰

VCS最大的挑戰就是管理**衝突 (conflict)**。

當有多個開發者同時對同一個檔案進行修改，而且VCS無法合併這些改變時就會發生衝突。

此時，只能手動解決衝突。





如何避免衝突?

- 模組化你的專案, 並讓不同的開發者開發不同的模組
- 不要commit專案自動產生的檔案(利用`.gitignore`)
- 經常整合
 - 整合其它人的版本
 - 分享自己改的部分
- 測試後再commit
- ...



VCS網頁服務

- GitHub
 - <https://github.com/>
- Gitlab
 - <https://gitlab.com>
- Bitbucket
 - <https://bitbucket.org>
- SourceForge
 - <http://sourceforge.net/>





免費工具

- Subversion (SVN)
 - Apache Subversion Server: <https://subversion.apache.org>
 - VisualSVN: <https://www.visualsvn.com>
 - TortoiseSVN Client: <https://tortoisesvn.net>
- Git
 - git-scm: <https://www.git-scm.com>
 - TortoiseGit Client: <https://tortoisegit.org>
 - Gitkraken: <https://www.gitkraken.com/>



Q & A



Computer History Museum, Mt. View, CA

