Migrate from TFS to Git (using Azure DevOps)

Roberson Liou



歡迎參加我們的每週四固定聚會

關於我

- 手沖咖啡科技宅
- 開發兼維運的偽全端工程師
- twMVC核心成員
- Blog：工程良田的小球場

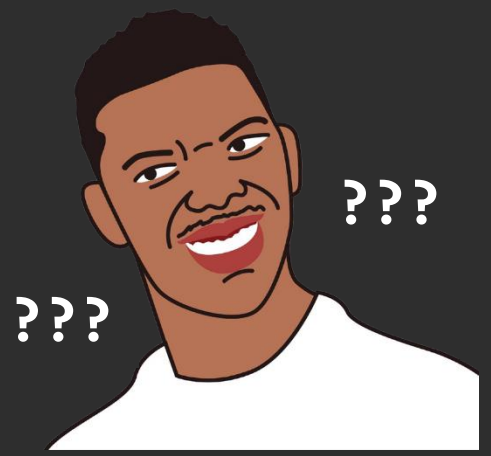


大綱

1. 移轉？吃飽沒事x
2. 分支策略選用
3. 歷史紀錄移轉
4. Azure DevOps With Git

tw MVC
https://mvc.tw

移轉？吃飽沒事x



為什麼要移轉?

Which version control system should I use?

Git is the default version control provider for new projects. You should use Git for version control in your projects unless you have a specific need for centralized version control features in TFVC.

- 分支成本(空間、時間)
- Pull Request
- 可離線的開發環境

tw MVC
https://mvc.tw

事前準備

- 決定使用工具
- 挑選分支策略
- 訂定歷史記錄移轉方式
- 舉辦Traning及Lab
- 等待適合時機進行移轉



使用工具



Visual Studio

- 地表最強的開發工具
- 整合Azure DevOps
- 介面支援多種Git操作









Azure DevOps Server

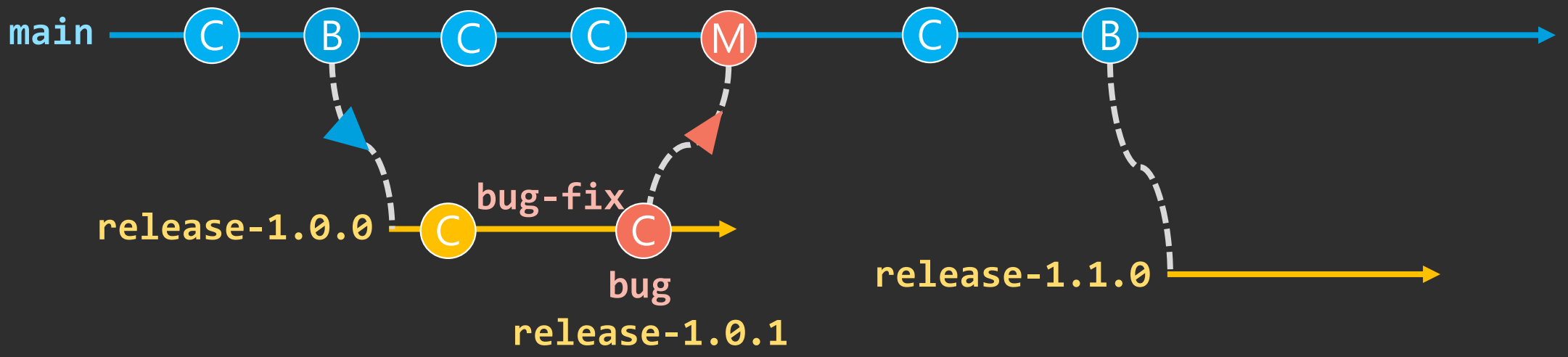


- 支援Git/TFVC
- 內建看板流程
- 整合CI/CD
- 自訂測試計畫

分支策略選用



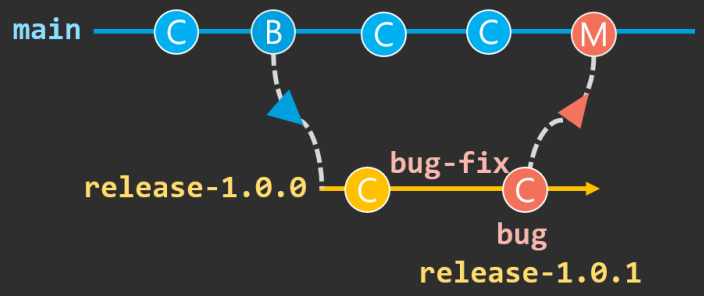
分支策略(TFS) - Release Isolation Flow



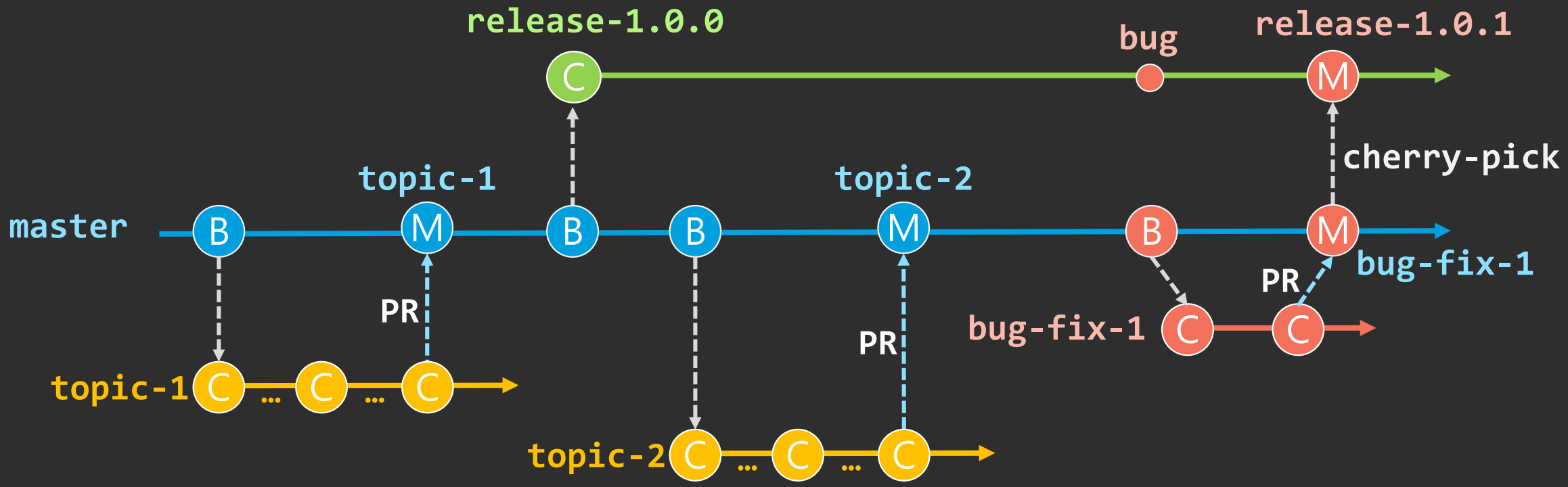
B ： 分支, Branch
C ： 提交, Commit
M： 合併, Merge

分支策略(TFS) - Release Isolation

- 優點
  - 分支策略簡單，學習曲線較低
  - 開發者僅需專注於main進行開發
  - Release分支獨立不受干擾
- 缺點
  - 主分支(main)穩定性不足
  - Release分支成本較高(package restore...)



分支策略(Git) - Trunk Based Development

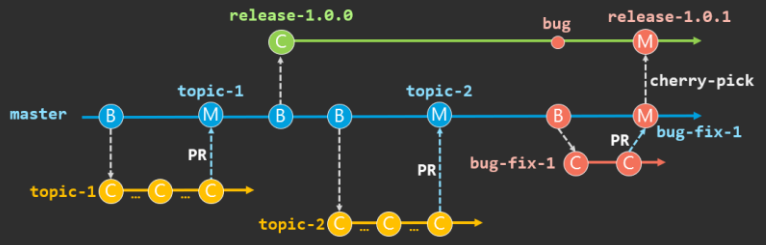


B ： 分支, Branch
C ： 提交, Commit
M： 合併, Merge

tw MVC
https://mvc.tw

分支策略(Git) - Trunk Based Development

- 優點
  - 分支週期短，反饋速度較快
  - 僅需維護一條master分支，減少合併衝突機率
- 缺點
  - 多條分支同時合併回master時，易發生建置資源搶用現象
  - topic分支容易受master分支影響



tw MVC
https://mvc.tw

歷史記錄移轉



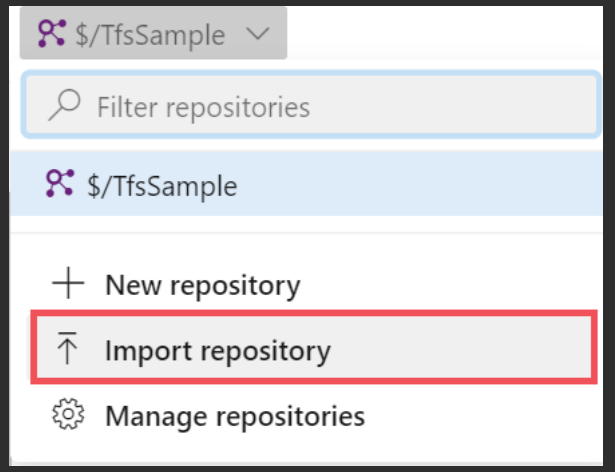
歷史紀錄該移轉嗎?

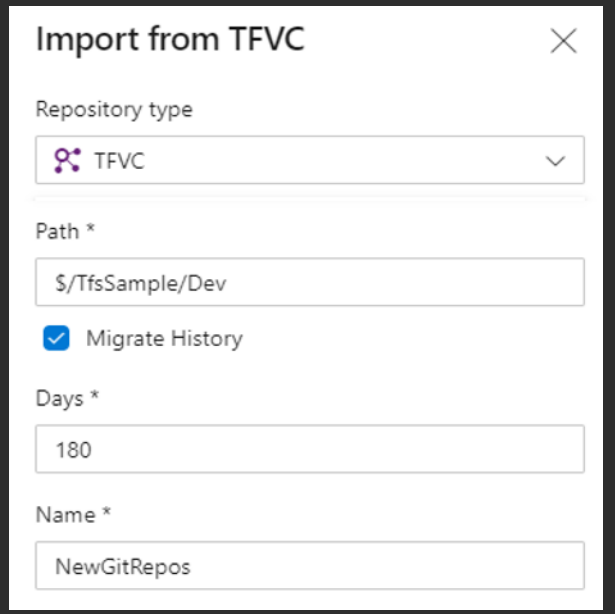
- TFS與Git先天不同
  - 分支方式
  - 檔案變更記錄儲存方式
- 「我們建議您不要移轉歷史記錄，而是只將最新的分支版本移轉至Git。」 – Matt Cooper, Program Manager, Azure DevOps
- 持續維護舊版控系統(TFS)
  - 於新版控系統的專案描述中放上舊版控的連結

tw MVC
https://mvc.tw

Azure DevOps內建移轉方式

- 僅支援Azure DevOps|TFS 2018|TFS 2017
- 兩個專案必須在同一個組織下
- 最多只能移轉180天的歷史記錄
- 專案大小不能超過1GB





git-tfs

- Open Source
- 提供git、tfs雙向移轉
- 指令式操作
  - `git tfs clone http://tfs:8080/tfs/DefaultCollection $/some_project`
- 移轉速度較慢
- 參考文章：TFVC - Git 專案搬家經驗談(黑暗執行緒)

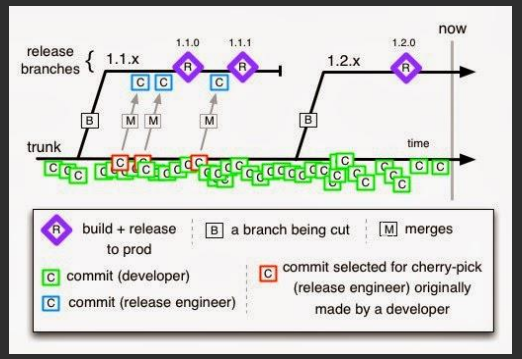
Training & Lab

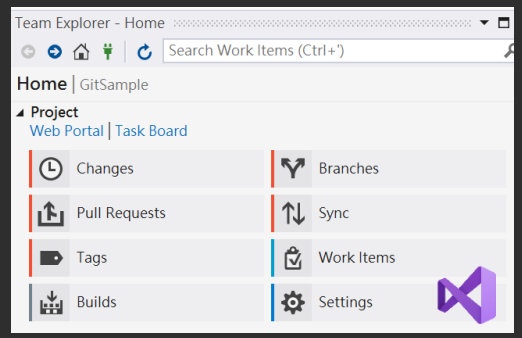


Training





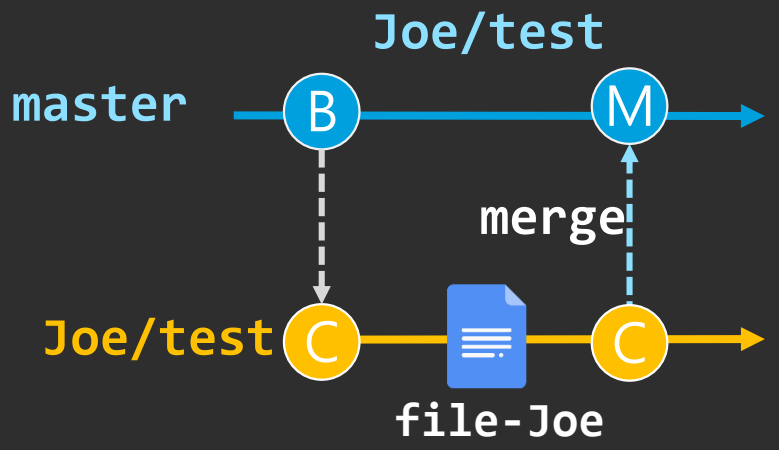


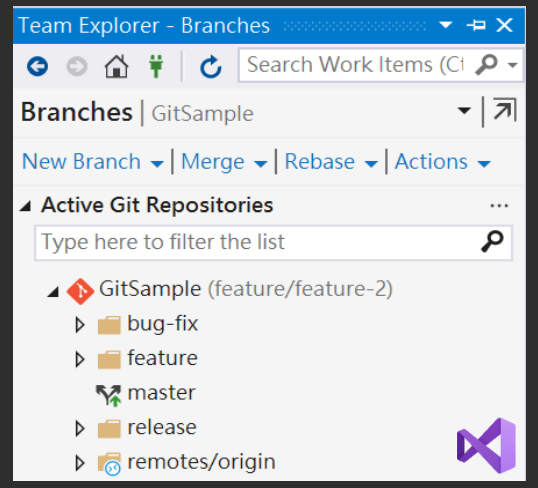


Lab



Azure DevOps

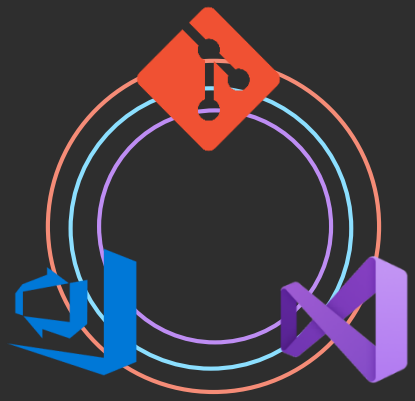




TFS & Git 指令比較

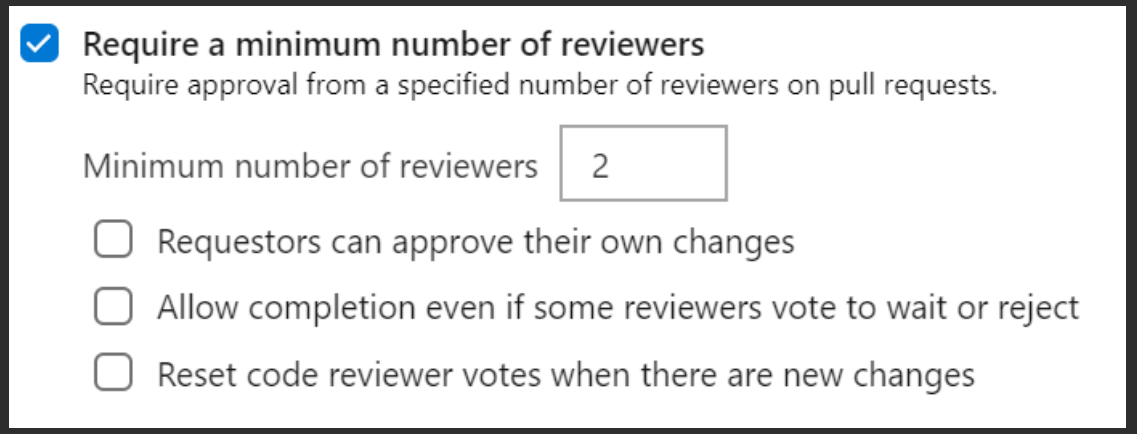
	TFS	Git
簽入(local)	X	commit
簽入(remote)	Check In	push
簽出	Get Latest Version	pull = fetch + merge
對照本機路徑	Map Local Path	clone
拆掉local commit	X	reset --hard
暫存程式碼	Shelveset	stash

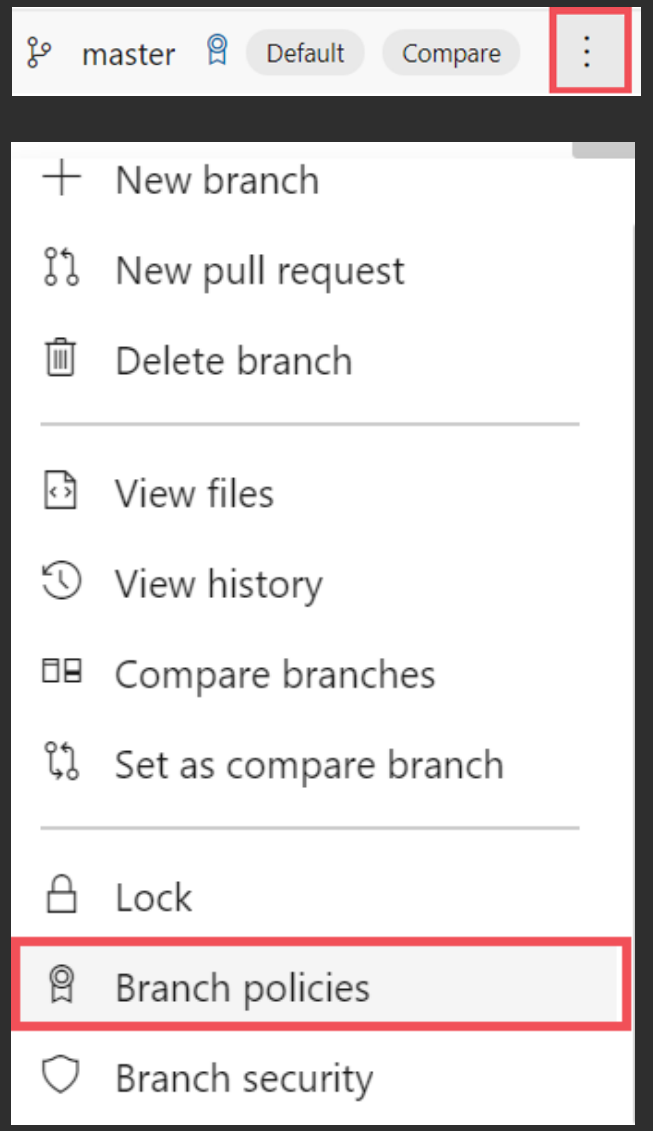
Azure DevOps with Git



Branch Policy(1)

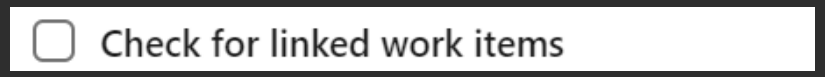
- 強迫該分支只接受PR
- 套用後該分支無法刪除
- 設定最低通過的Reviewer人數



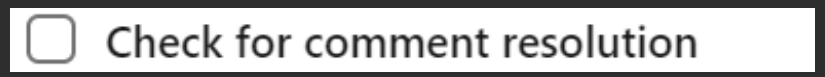


Branch Policy(2)

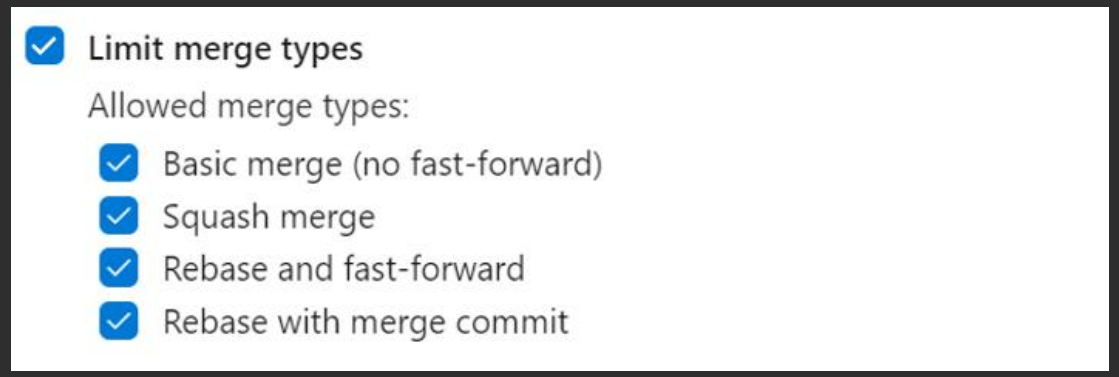
- 檢查是否有工作項目連結



- 確認每個回覆是否有解決



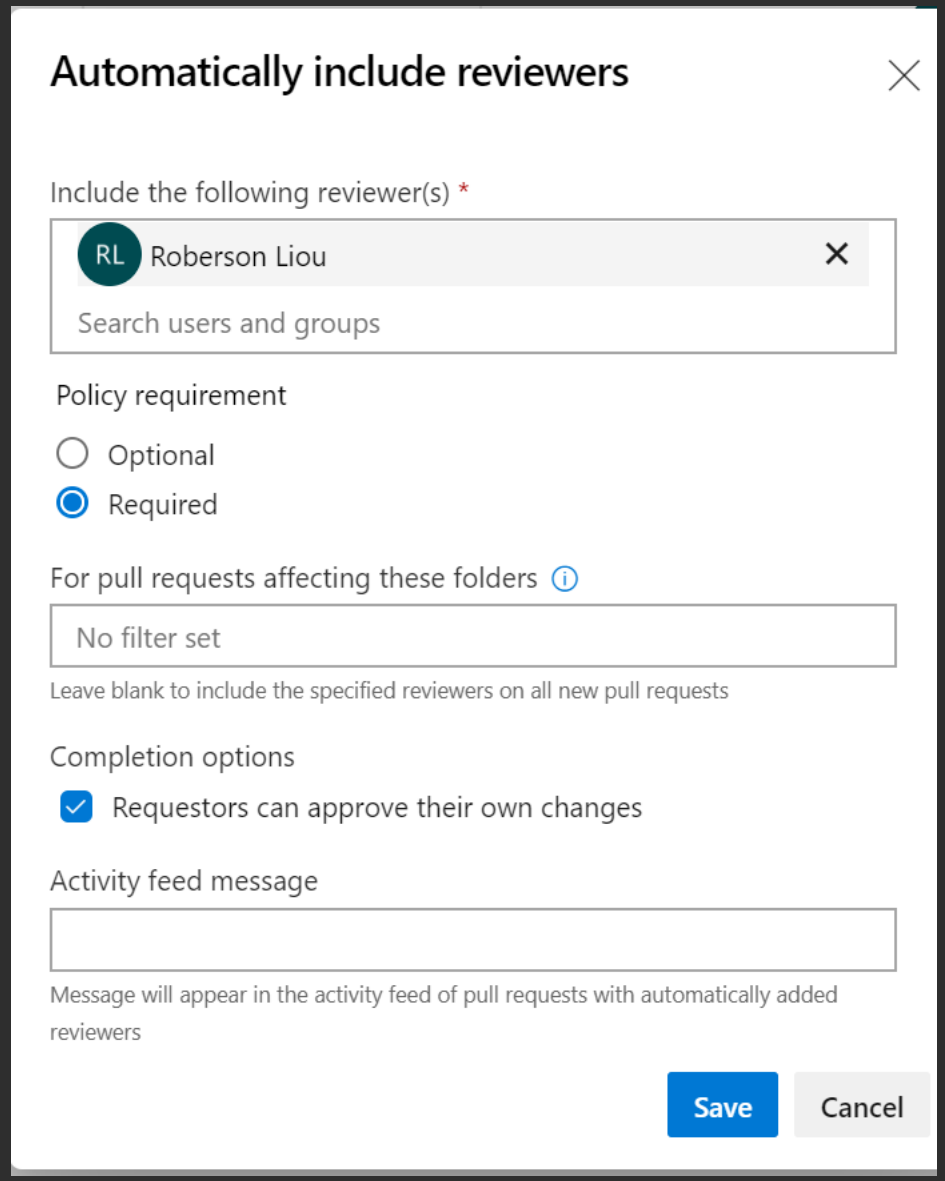
- 限制合併方式



tw MVC
https://mvc.tw

Branch Policy(3)

- 驗證建置及測試是否通過
  - 須先建立CI Pipeline
- 設定預設Reviewer名單
  - 可指定群組

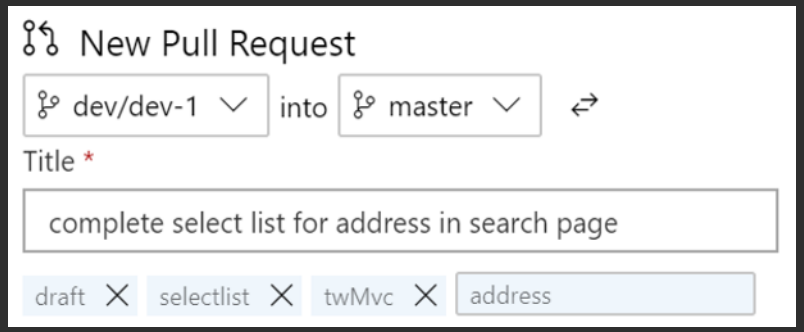


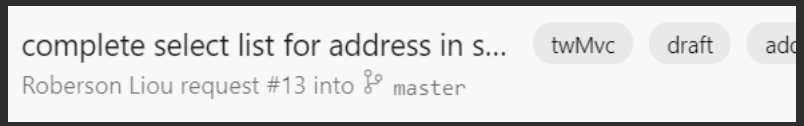
Pull Request

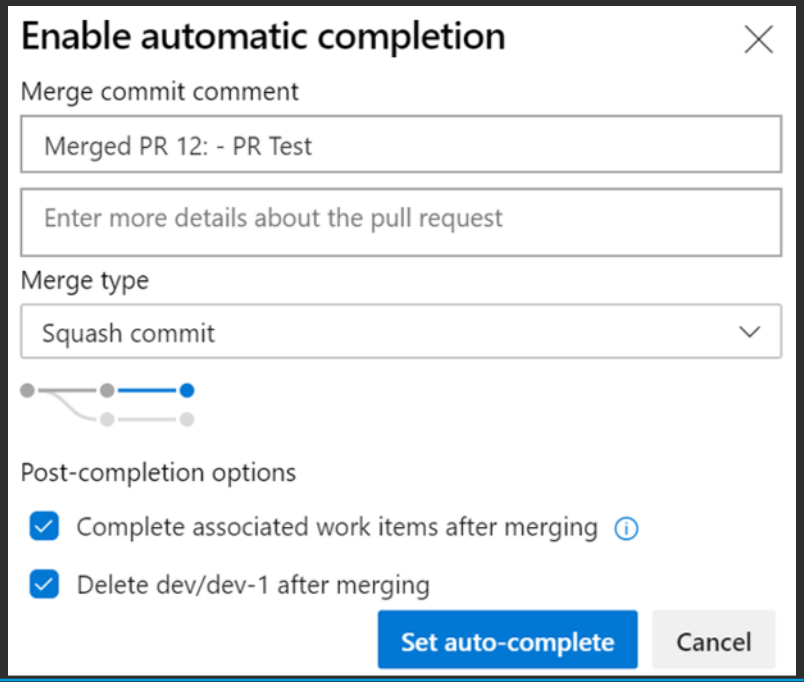
- 協助程式碼審查流程



- 可加入自訂標籤
- 在PR完成時...
  - 可挑選合併方式
  - 可在合併後刪除分支
  - 可自動完成連結的工作項目

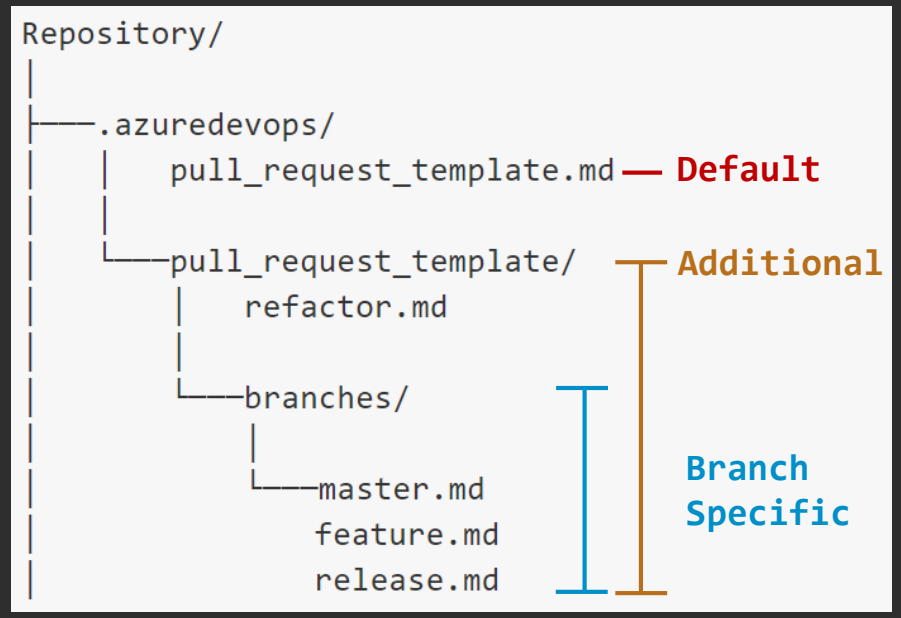


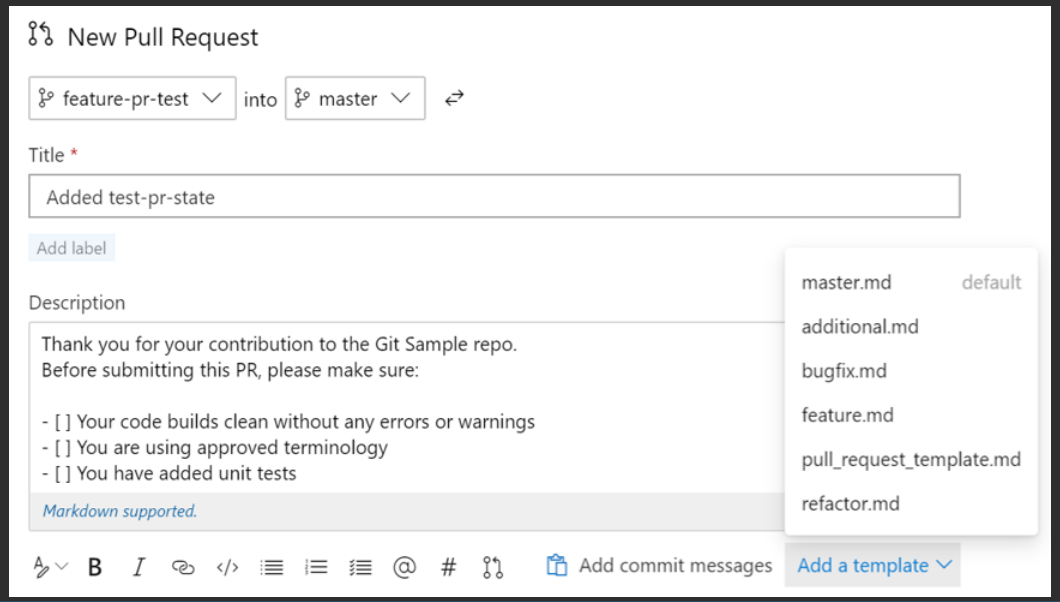




Pull Request Template

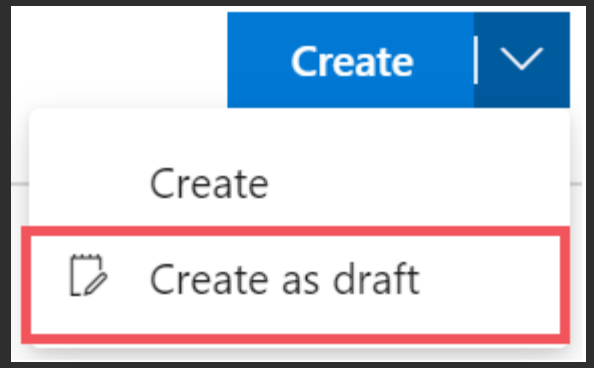
- 可引導Requestor描述情境
- 套用方式分為3種
  - Default
  - Branch Specific
  - Addtional
- 命名參考(MSDN)

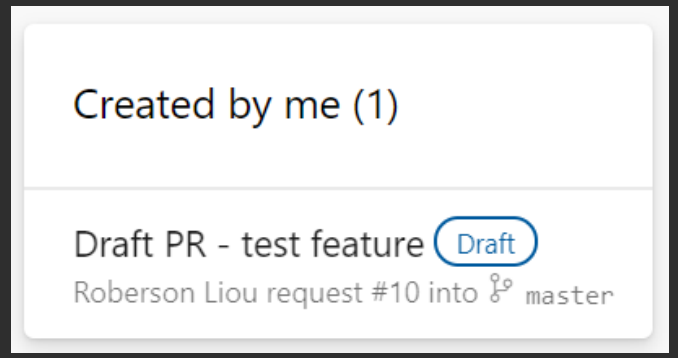




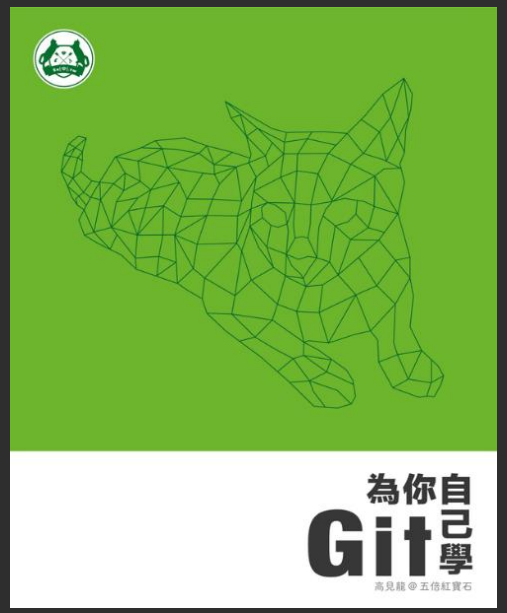
程式還沒寫完但想找人Review?

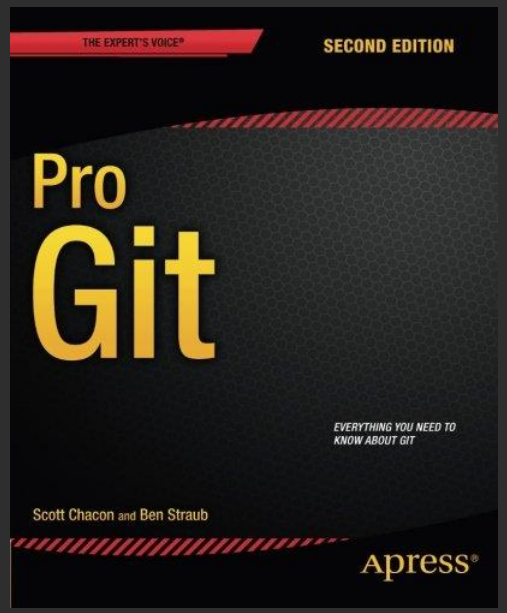
- 走一般PR機制可能會不小心被合併
- 標題開頭加上「WIP」、「Do-not-Merge」
- Draft Pull Request
  - 不會觸發Build Validation
  - 預設的Reviewer不會被加入
  - 當功能完成後可直接切換為Publish模式

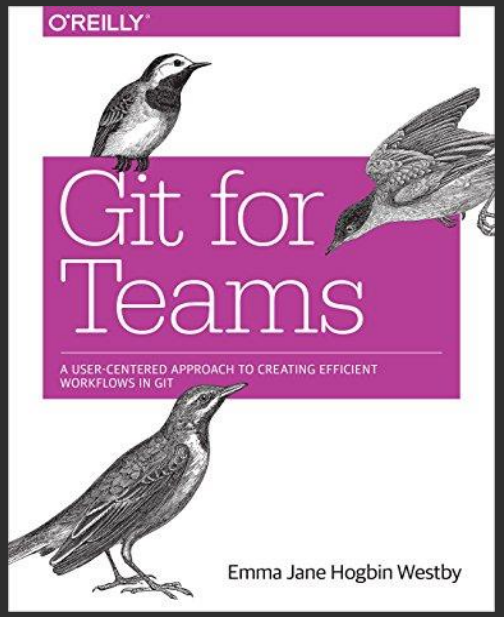




如果你想學Git...


為你自己學
Git
高見龍@五倍紅寶石


THE EXPERT'S VOICE®
SECOND EDITION
Pro
Git
EVERYTHING YOU NEED TO KNOW ABOUT GIT
Scott Chacon and Ben Straub
Apress®


O'REILLY®
Git for Teams
A USER-CENTERED APPROACH TO CREATING EFFICIENT WORKFLOWS IN GIT
Emma Jane Hogbin Westby

推薦閱讀 – Git & Branch Strategies

- Git Turtorials(Bitbucket)
- Branch Strategies for TFS(MSDN)
- Trunk Based Introduction
- How We Use Git At Microsoft(MSDN)



推薦閱讀 – How to Migrate

- 從集中版本控制到分散式版本控制(王建興)
- TFS Git 筆記 - 該用 TFVC 還是 Git？(黑暗執行緒)
- TFS Git 筆記 - TFVC - Git 專案搬家經驗談(黑暗執行緒)
- Map TFVC actions to Git(MSDN)
- Plan your migration to Git(MSDN)

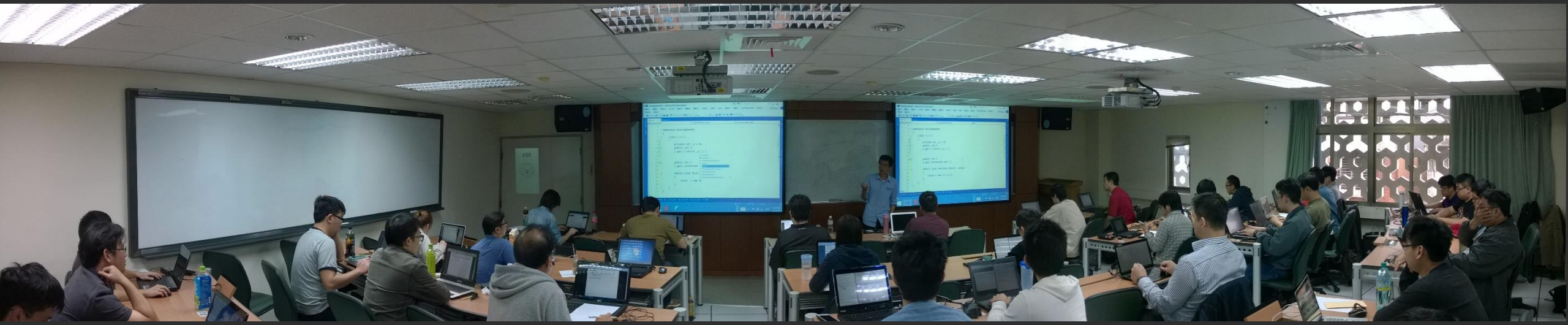


Blog 是記錄知識的最佳平台



https://dotblogs.com.tw

SkillTree 為了確保內容與實務不會脫節，我們都是聘請企業顧問等級並且目前依然在職場的業界講師，我們不把時間浪費在述說歷史與沿革，我們並不是教您考取證照，而是教您如何上場殺敵，拳拳到肉的內容才是您花錢想要聽到的，而這也剛好是我們擅長的。



https://skilltree.my

天瓏資訊圖書

