儲存庫持續同步之後：遷移切換該如何驗收？

2026-11-24 · 8plus

GitHub 在 2026 年 11 月 19 日宣布，Azure DevOps Services 到具資料落地區域選項的 GitHub Enterprise Cloud，其 Enterprise Live Migrations 已正式推出。持續同步讓團隊可以把最後切換安排在選定時間；但「同步完成」與「可以交付」仍是不同的驗收問題。

先說清楚搬了什麼、留下什麼

公告指出，遷移涵蓋 Git 歷史與 PR 等內容，分支政策會轉成 rulesets；Azure Boards 工作項目與 Pipelines 定義仍留在 Azure DevOps。這不是把整個研發系統原封不動搬到新平台。

以下是 8plus 提出的交付檢查方式，不是已實測的遷移案例，也不是對任何專案保證零停機。產品適用條件與操作限制，仍應依官方文件及專案環境確認。

驗收單位應該是一條工作路徑

假設團隊每天的路徑是：從工作項目開分支、送出 PR、等待檢查、合併、部署。若只驗證新平台可以 clone，就只確認了其中一小段。比較有用的驗收，是選一個不含敏感資料的測試變更，把這條路徑走完。

每一步都留下可核對的結果：預期分支是否存在、測試提交是否相同、必要檢查是否真的阻擋未通過的合併、部署是否指向預期環境。不要為了完成演示，暫時繞過原本應該存在的保護。

持續同步期間，仍需要一份切換約定

同步減少了等待搬檔案的時間，卻不能代替團隊決定「從何時起在哪裡工作」。切換前要約定最後一次來源核對、停止舊入口寫入的時點、新入口開放條件，以及誰有權宣布完成。

例如，可以把最後核對的來源提交、目標提交與檢查時間記在同一張交接單。若來源仍有新寫入，就重新確認差異，不以幾分鐘前的截圖當作最新證據。這份紀錄的目的，是讓下一位值班者知道哪個狀態才是驗收依據。

把留在原平台的依賴列出來

工作項目與 pipeline 定義仍留在原處時，新儲存庫只是工作流程的一部分。應另外確認連線權限、觸發條件、服務身分與部署目的地；不能因為 PR 歷史看起來完整，就推論部署也已經正常。

驗收紀錄可以只保留系統名稱、負責人角色與檢查結果。憑證內容、客戶資料與私人帳號資訊，不應複製到公開 issue 或交付文件。若某一條連線還未驗證，就直接列為未完成，不用「大致正常」掩蓋它。

回復方案要回答新寫入如何處理

切換後若發現問題，重新指回舊網址不一定能恢復工作。新平台上可能已產生新的提交或討論。回復方案應先說明哪些寫入需要保存、由誰比對，以及何時停止繼續切換，避免兩邊同時成為大家認為的正式入口。

可以在演練環境刻意讓一個必要檢查失敗，觀察團隊是否知道要停在哪一步。這個練習驗證的是操作約定與責任分工，不是用破壞正式環境來證明回復能力。

交付完成，是下一個人能接著工作

一份實用的遷移交接，可以用五個答案收尾：正式入口在哪裡、最後核對的是哪個狀態、哪條工作路徑已通過、哪些依賴仍待處理、遇到問題找誰。工具讓資料移動更順，團隊仍要讓工作接續得清楚。

來源與延伸閱讀

- GitHub 官方公告（2026-11-19）
- 多帳號交付前的身分核對
- PR 事件通知後的狀態核對