HƯỚNG DẪN SỬ DỤNG TORTOISESVN + GITHUB

(Yêu cầu: phải cài TortoiseSVN)

# Cấu trúc Github

* Branches: Thư mục mặc định của GITHUB, không cần quan tâm
* Trunk: Chứa các thư mục chính của dự án trên Github
  + Finals: Thư mục chứa các file sản phẩm của dự án, không commit lên đây khi không có sự đồng ý của PM
  + References: Các file tài liệu liên quan để tham khảo, không commit lên đây khi không có sự đồng ý của PM
  + Work In Progress: Khu vực làm việc của các thành viên, commit nộp công việc của mình ở đây
    - Coding: Source Code của dự án
    - Document: Các tài liệu của dự án đang trong giai đoạn chỉnh sửa, chưa được sự chấp thuận của các thành viên để lên finals
    - Member’s Workspace: Các phần việc của thành viên khi chưa ghép vào document hay coding

# Chú ý

* Nên tạo một folder chứa công việc riêng, sau khi làm thấy hoàn chỉnh mới commit lên GITHUB, không nên commit liên tục lên GITHUB, sẽ khó cho những người khác khi cập nhật lên không kịp.
* Nên update thư mục trước khi commit bất kỳ điều gì để đảm bảo việc commit xảy ra ổn và không xung đột, gây lỗi GITHUB
* Khi commit nên có message cho người khác dễ tìm hiểu
* Không commit với bất kỳ một thư mục nào rỗng, sẽ không commit được
* Commit tại thư mục ngoài cùng, chứa branches và trunk, commit ở thư mục bên trong sẽ chỉ commit những thay đổi trong thư mục đó mà không có các thư mục cùng cấp.

# Quy tắc đặt tên

## Với các tài liệu Doc

* Với các file trong Work In Progress:
  + Tên file được đặt theo quy tắc [Tên tài liệu]\_[Tên người làm]\_[Version]\_[Người Review(nếu có]

Ví dụ: Research Các đối thủ cạnh tranh\_VuPT\_v1.0.docx

* Với các tài liệu trong Final
  + Đặt tên theo [Tên đề tài]\_[Tên tài liệu]\_[Version]
* Với các tài liệu trong References
  + [Lĩnh vực liên quan]\_[Tên tài liệu]\_[Tên người làm]
  + Ví dụ: Tutorial\_Hướng dẫn sử dụng TortoiseSVN và Github\_VuPT.docx

## Với các tài liệu Code

* Đặt tên theo Coding Convention của từng loại Code tương ứng.

## Version được tính như sau:

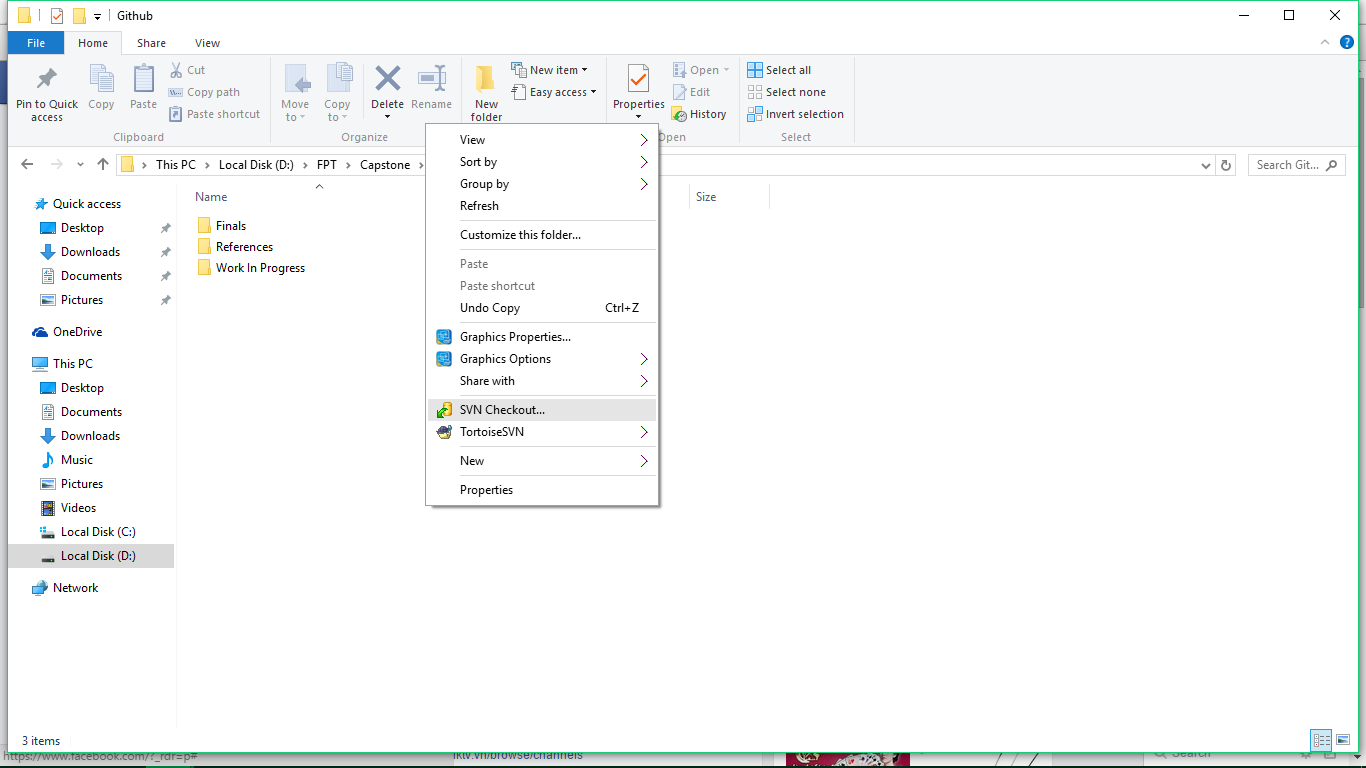
* Bản đầu tiên là 1.0
* Các bản cập nhật nhỏ thì lên 1.0a
* Cập nhật nhiều thì 1.1
* Sửa đổi lớn thì lên 2.0

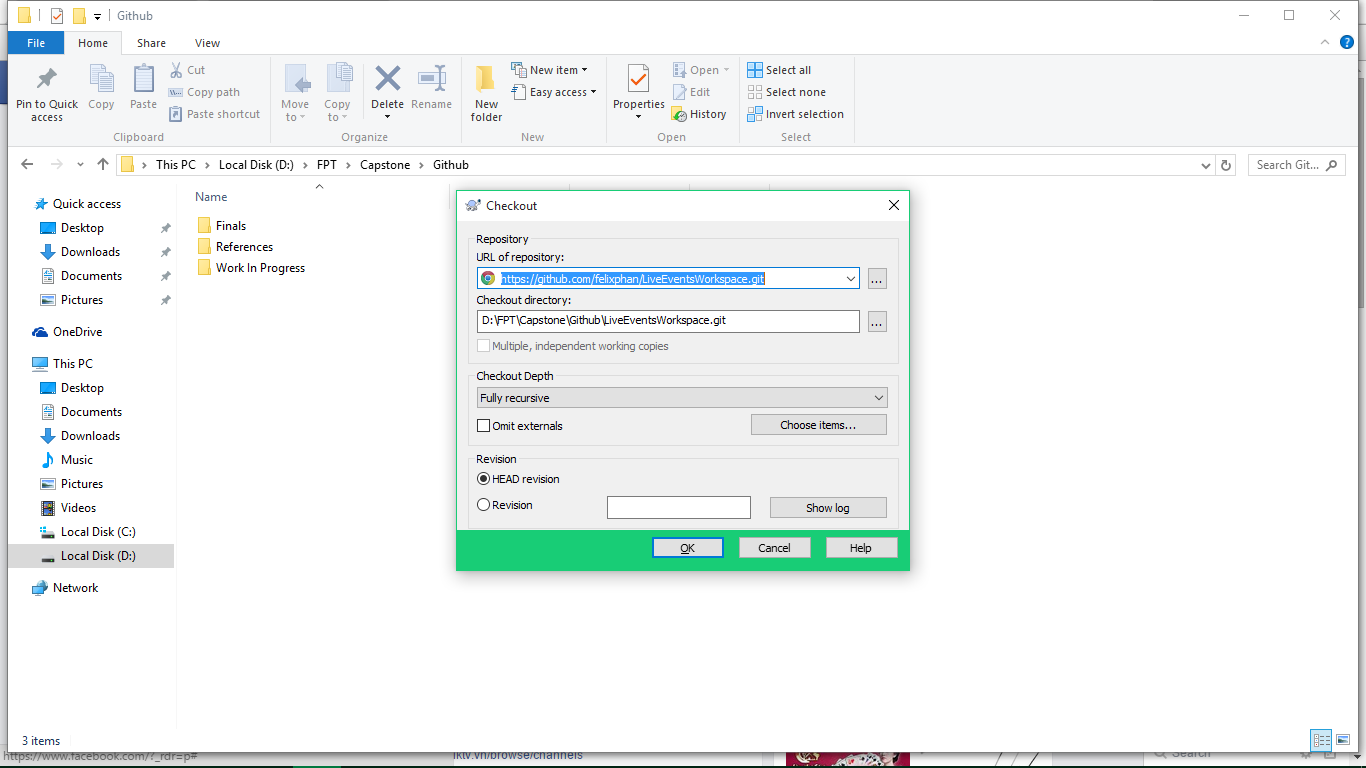
# Checkout

* Checkout là cách lấy dữ liệu trên Github xuống LẦN ĐẦU TIÊN cho một thư mục.

B1: Tạo một thư mục mới

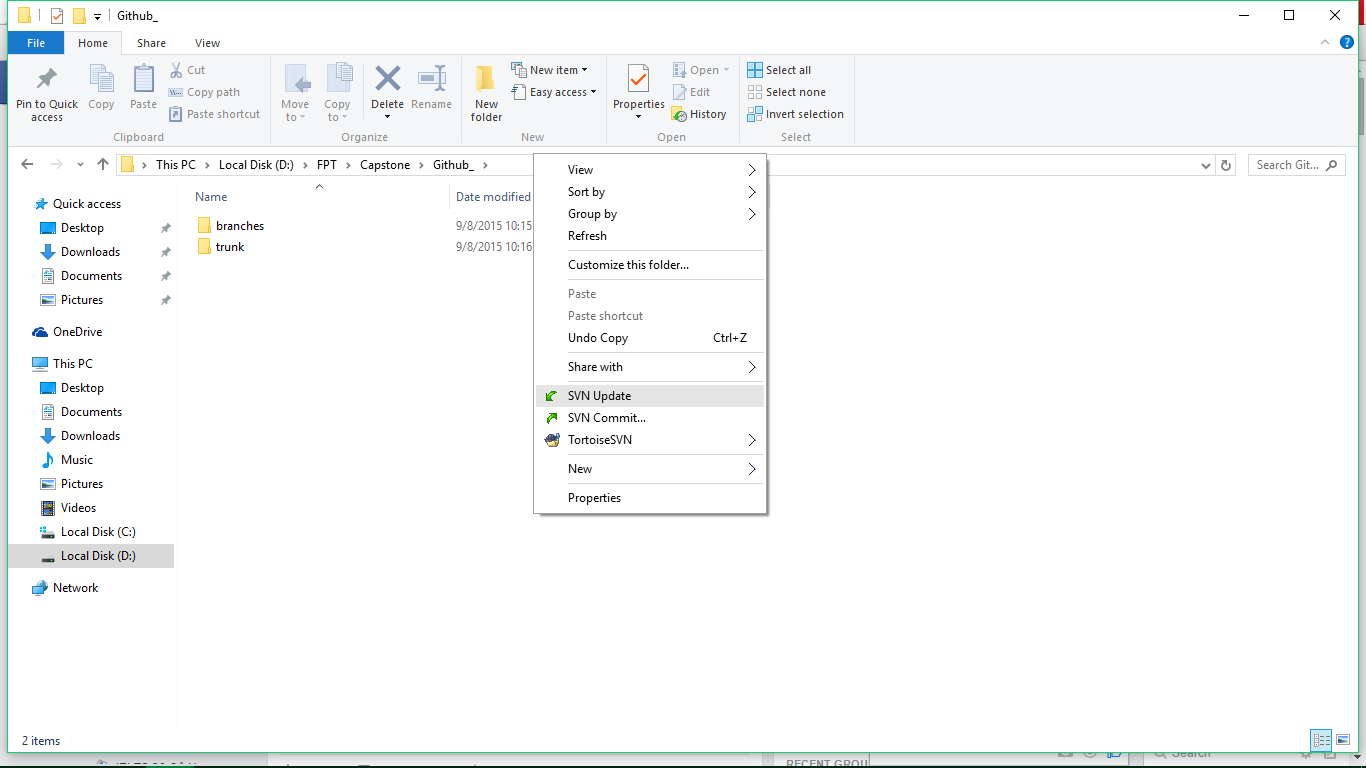
B2: Vào bên trong thư mục đó, chuột phải chọn Checkout



B3: Nhập đường link thư mục Github vào trong URL

# Update

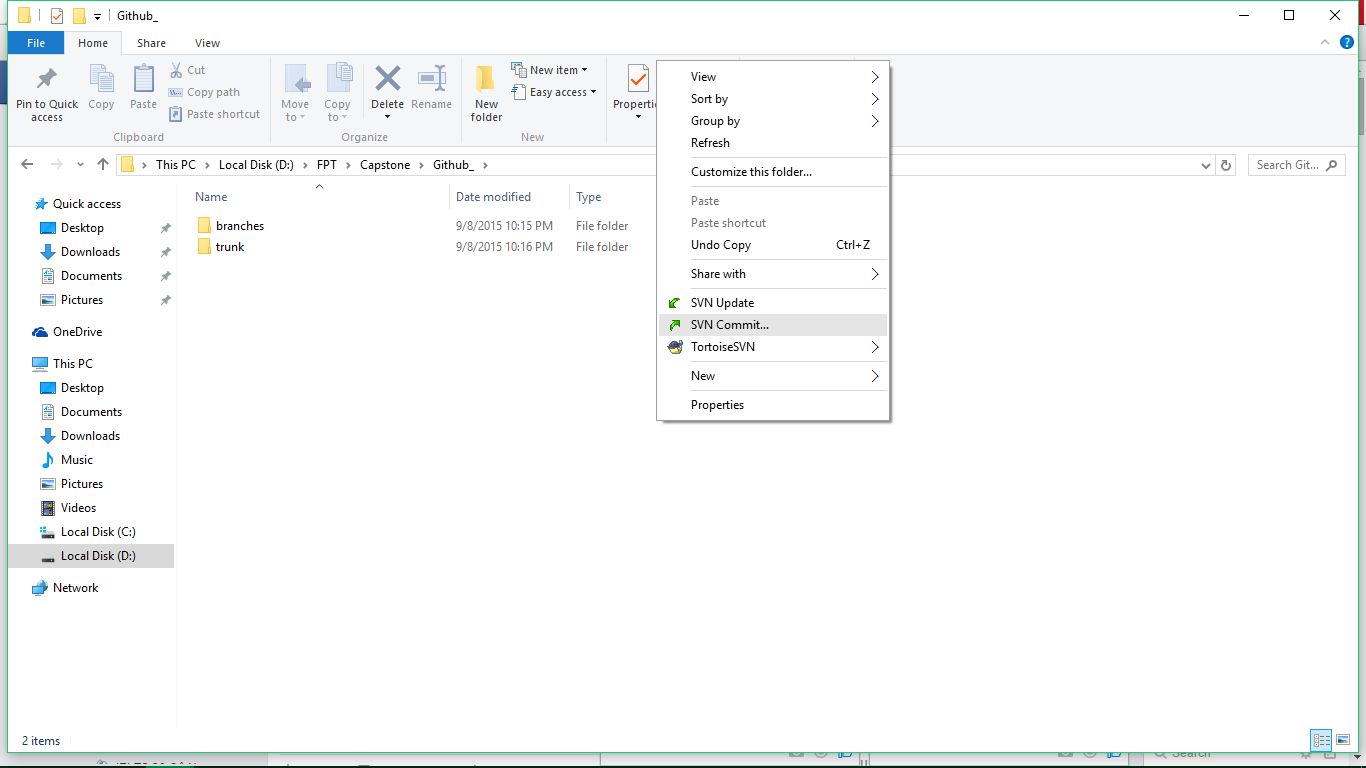
* Cập nhật thư mục ĐÃ ĐƯỢC CHECKOUT

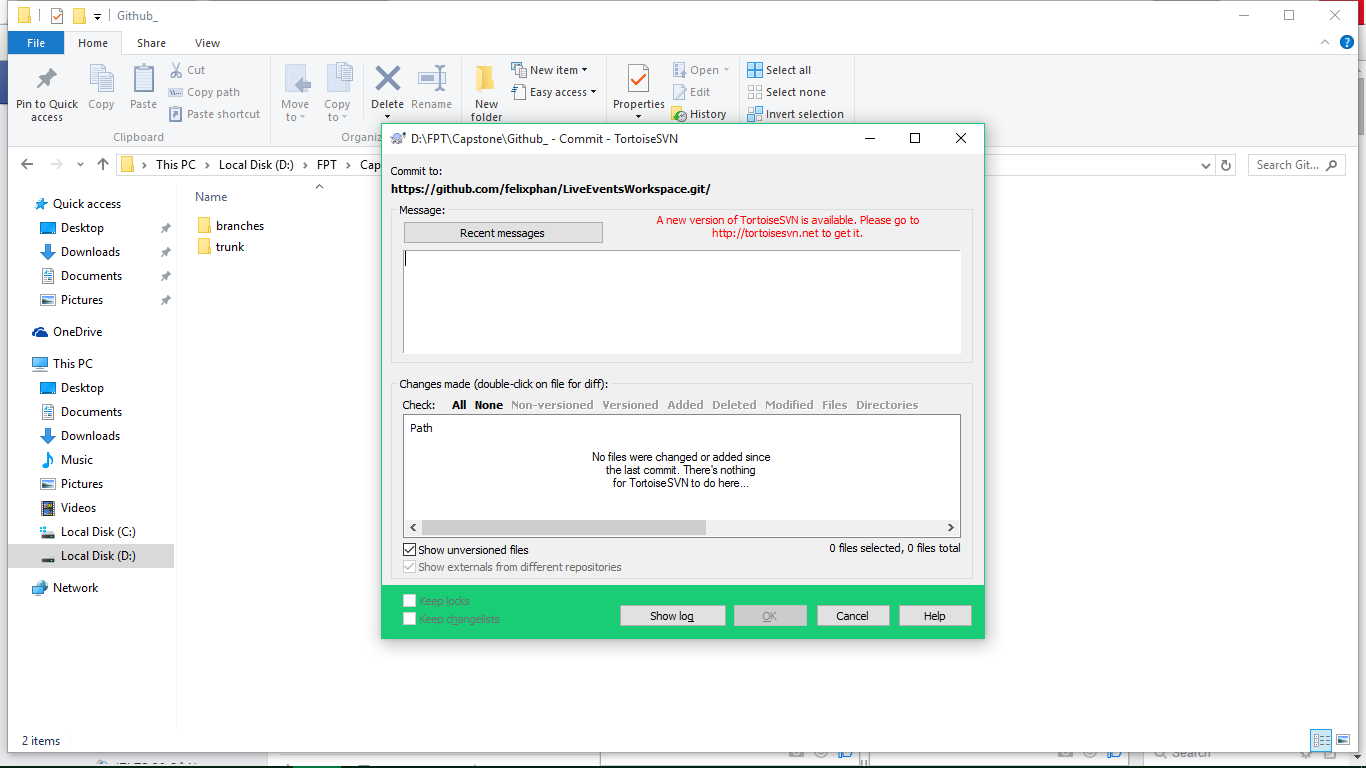


* Luôn phải thực hiện bước này trước khi Commit bất cứ file nào lên GITHUB

# Commit

* Đưa một hoặc nhiều file lên GITHUB





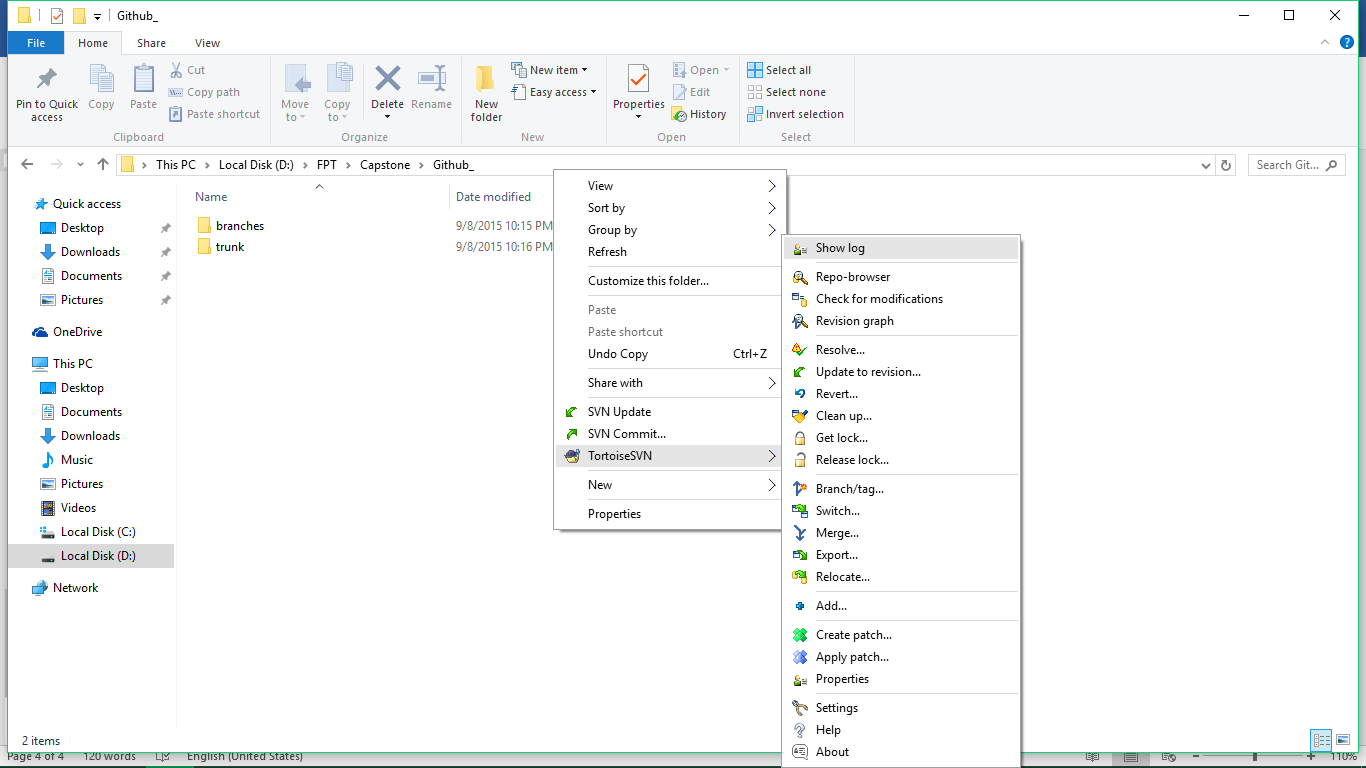
Chọn Check All

NGHIÊM CẤM việc Commit mà không ghi Message.

Message là nội dung mình Commit lên bao gồm những gì, có nội dung gì

# Show Log

* Kiểm tra xem những người khác đã commit những gì



# Update to Revision

* Trường hợp update hoặc đang ở phiên bản lỗi, có thể chọn update trở về 1 revision nào đó.

