**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

**KHOA CÔNG NGHỆ THÔNG TIN**

\*\*\*\*\*

****

**BÁO CÁO**

**Môn: Thực tập nhóm**

Nhóm : 2

Giáo viên hướng dẫn : Phan Nguyên Hải

Các thành viên: Nguyễn Tuấn Hiệp

Tạ Quỳnh Anh

Lê Thị Hương

Nguyễn Thị Hương

Đặng Thị Dung

**Hà Nội, 2018**

# 1. Tìm hiểu về Subversion (SVN)

**"Subversion" (viết tắt SVN)**là **"một hệ thống quản lý version (version control system - VCS) được giới thiệu vào năm 2000 bởi công ty CollabNet (http://subversion.tigris.org). Đây là hệ thống hỗ trợ làm việc theo nhóm rất hiệu quả"**.  
  
**Phần mềm:**  
  
**Cho client:** TortoiseSVN, **Download:http://tortoisesvn.net/**  
  
**Cho server:** VisualSVN – Server **Download:** <http://tortoisesvn.net/downloads.html>

## 1.1 Chức năng Subversion

**Khi một nhóm làm việc trên cùng một project, việc nhiều người cùng chỉnh sửa nội dung của một file là điều không thể tránh khỏi. SVN Subversion cung cấp các chức năng để có thể thực hiện việc này một cách đơn giản và an toàn.**

* **SVN Subversion** được thiết kế với mục đích thay thế hệ thống quản lý phiên bản Concurrent Versioning System (CVS) đã cũ và có nhiều nhược điểm. **Subversion** có thể được sử dụng để quản lý bất cứ hệ thống phiên bản nào.
* **SVN Subversion** là hệ thống quản lý source code tập trung (Centralized).
* **SVN Subversion**là hệ thống quản lý phiên bản mạnh mẽ, hữu dụng, và linh hoạt.
* **SVN Subversion** quản lý tập tin và thư mục theo thời gian.
* **SVN Subversion** giống như một hệ thống file server mà các client có thể download và upload file một cách bình thường.
* Điểm đặt biệt của **SVN Subversion** là nó lưu lại tất cả những gì thay đổi trên hệ thống file: file nào đã bị thay đổi lúc nào, thay đổi như thế nào, và ai đã thay đổi nó.
* **SVN Subversion** cũng cho phép recover lại những version cũ một cách chính xác. Các chức năng này giúp cho việc làm việc nhóm trở nên hiệu quả và an toàn hơn rất nhiều.

Thông thường, client và server kết nối thông qua mạng LAN hoặc Internet. Client và server có thể cùng chạy trên một máy nếu**SVN Subversion** có nhiệm vụ theo vết lịch sử của dự án do các nhà phát triển phần mềm phát triển trong nội bộ.  
  
**SVN Subversion**hỗ trợ khá nhiều giao thức để kết nối giữa client và server.  
  
Ví dụ bạn có thể dùng các giao thức của ứng dụng web như http:// hoặc https://, hay các giao thức của svn như svn:// hoặc svn+ssh://, hoặc nếu phần mềm client và server cài chung trên 1 máy thì có thể dùng file://.  
Việc cho phép server hỗ trợ giao thức nào phụ thuộc vào lúc cấu hình.

* **Cài đặt SVN Subversion** (Client): tool dùng trên Client
* Cài đặt VisualSVN(Server): tool dùng cho Server

Chi tiết cài đặt và cấu hình cho Server và Client (Đầy đủ hướng dẫn từ cài đặt đến cách sử dụng)

**Kết Luận: "Subversion" (viết tắt SVN)**là **"một hệ thống quản lý version (version control system - VCS) được giới thiệu vào năm 2000 bởi công ty CollabNet (http://subversion.tigris.org). Đây là hệ thống hỗ trợ làm việc theo nhóm rất hiệu quả"**.

## Cài đặt SVN cho Client

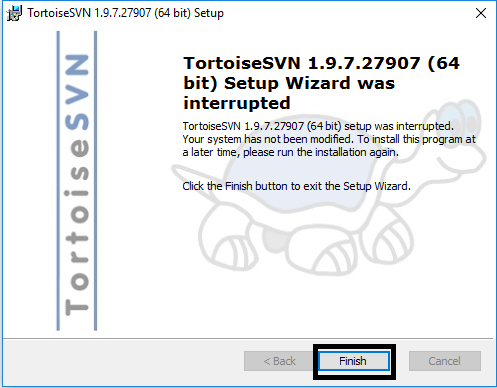
* Download tại trang <https://tortoisesvn.net/downloads.html>



* Double click vào file để cài -> Next

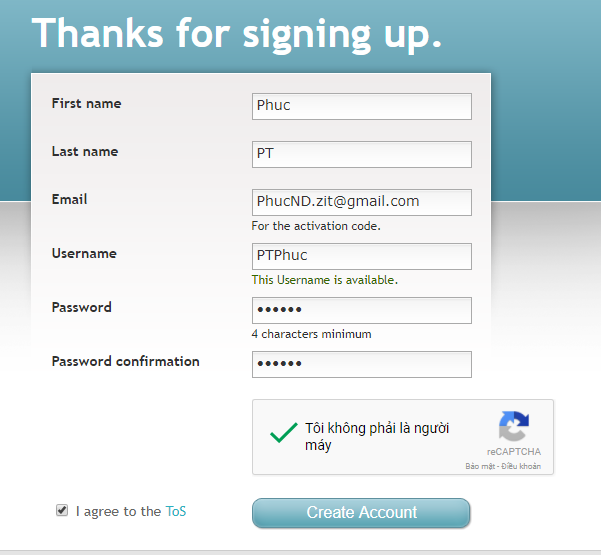


* Chọn vị trí cài đặt -> Next
* Click Install để cài đặt
* Quá trình cài đang thực hiện
* Cài đặt hoàn thành -> Finish

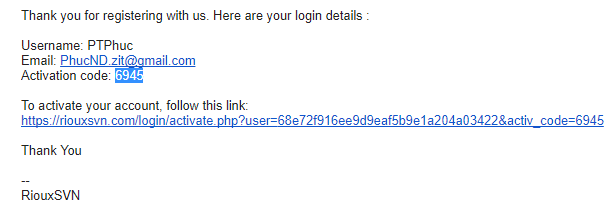


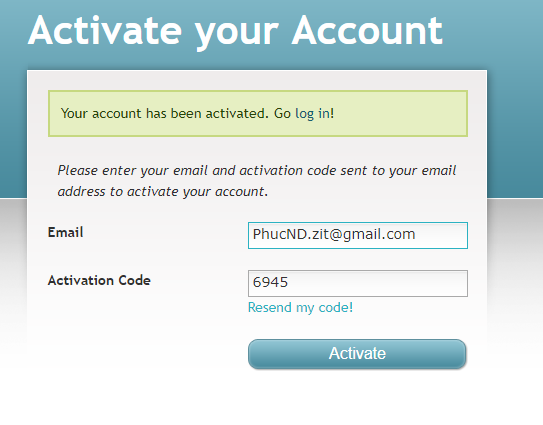
## Cách tạo tài khoản

* Vào link: <https://riouxsvn.com/register>
* Điền Đủ thông tin vào form gồm :
* Fist Name
* Last name
* Email
* Username
* Password
* Password confirmation
* CapCha
* Đồng ý với các điều khoản



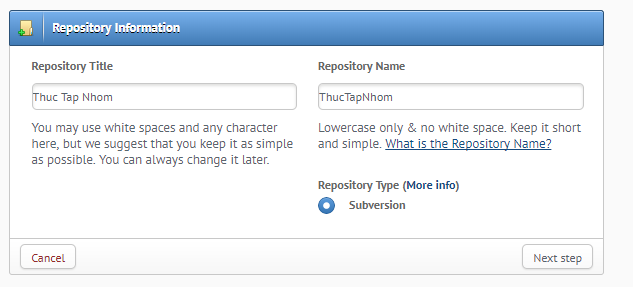
* Activate your account





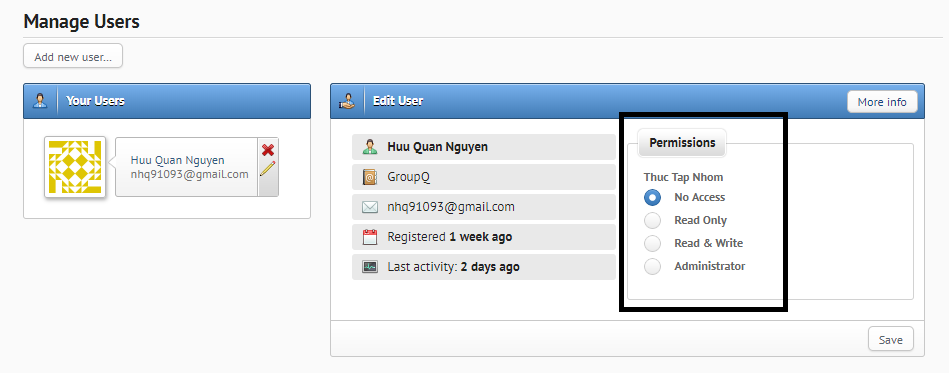
## Create a New Repository

* Vào tab Repository click Create new repository…



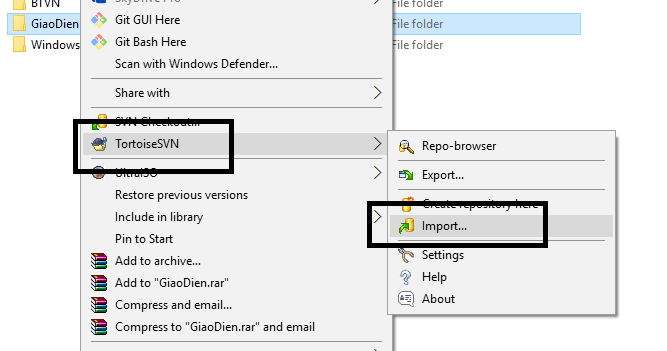
## Phân Quyền

* Vào tab User click vào Tên User cần phân quyền

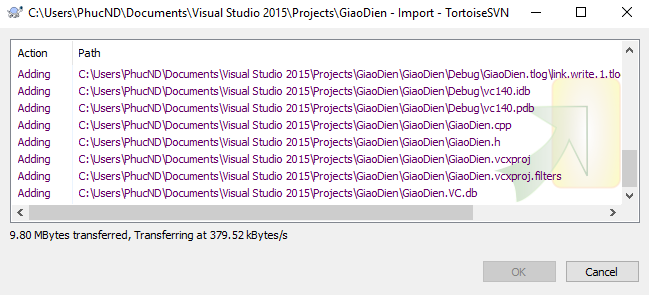


## Up Code lên

* Chuột Phải chọn TortoiseSVN Chọn Import

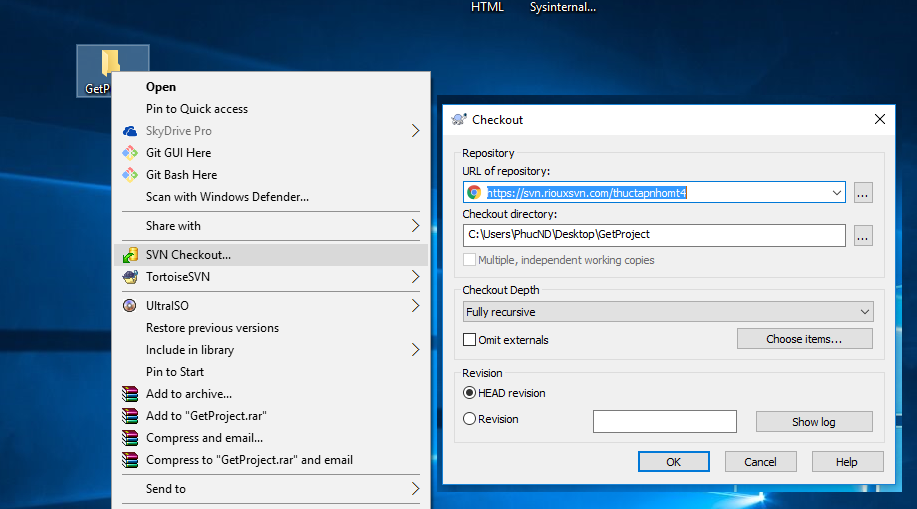


* Click và đợi tải lên host

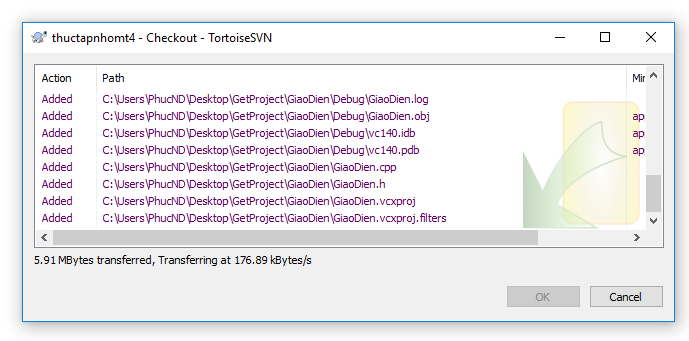


## Lấy dự án về

* Chuột Phải chọn SVN Checkout

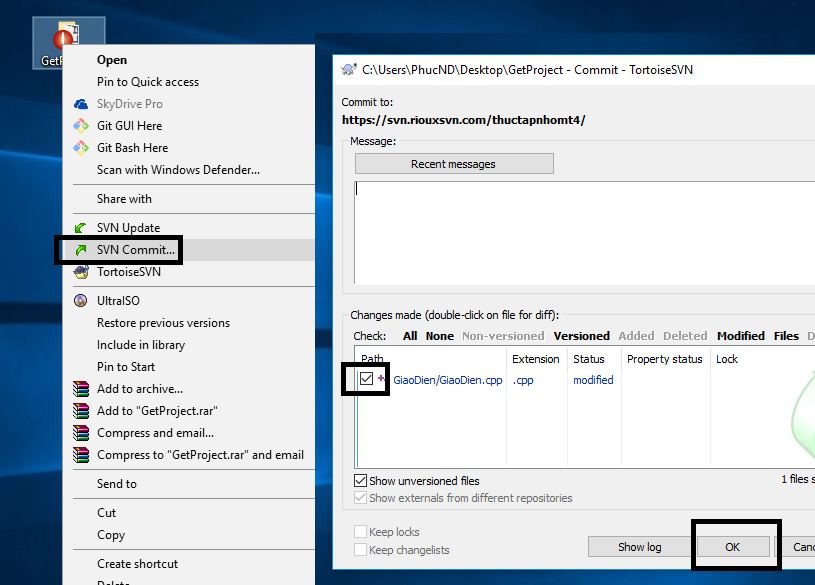


* Nhấn Ok và đợi tải về



## Commit sau chỉnh sửa

* Click vào Forder Chọn SVN commit và ấn Ok (add Message ghi chú)

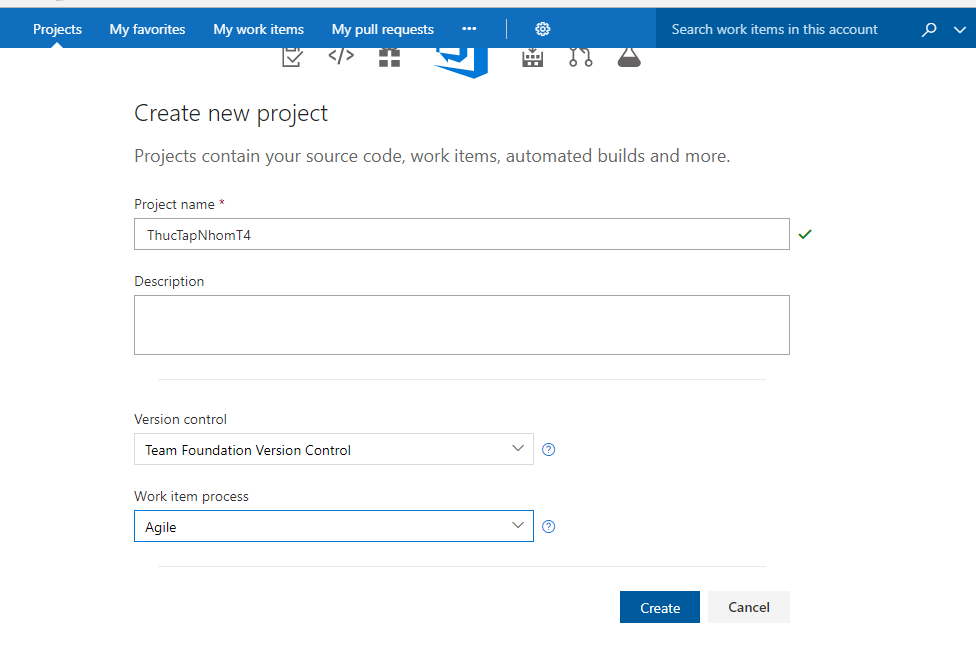


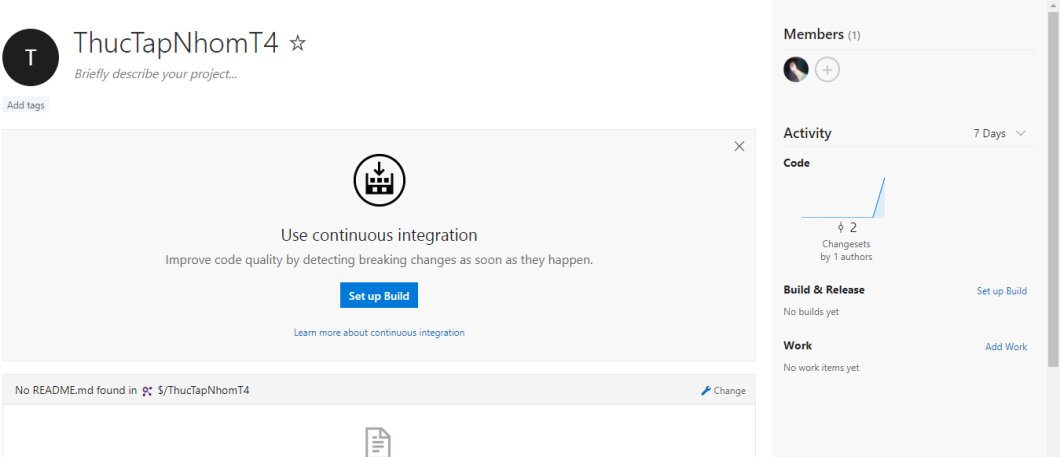
# 2. Triển khai Phần mềm quản lý phiên bản trên mạng máy tính (TFS)

## 2.0 Giới Thiệu

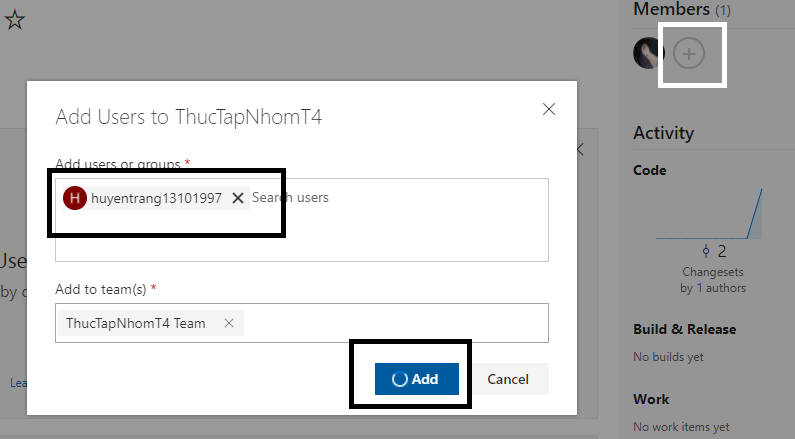
* **Team Foundation Server** (**TFS**) **là** một chương trình server được sử dụng để quản lý mã nguồn của các lập trình viên trong các dự án chung. ... Lưu trữ các phiên bản của mã nguồn và cho phép tải lại khi cần. Quản lý thay đổi trong project

## 2.1 Tạo project



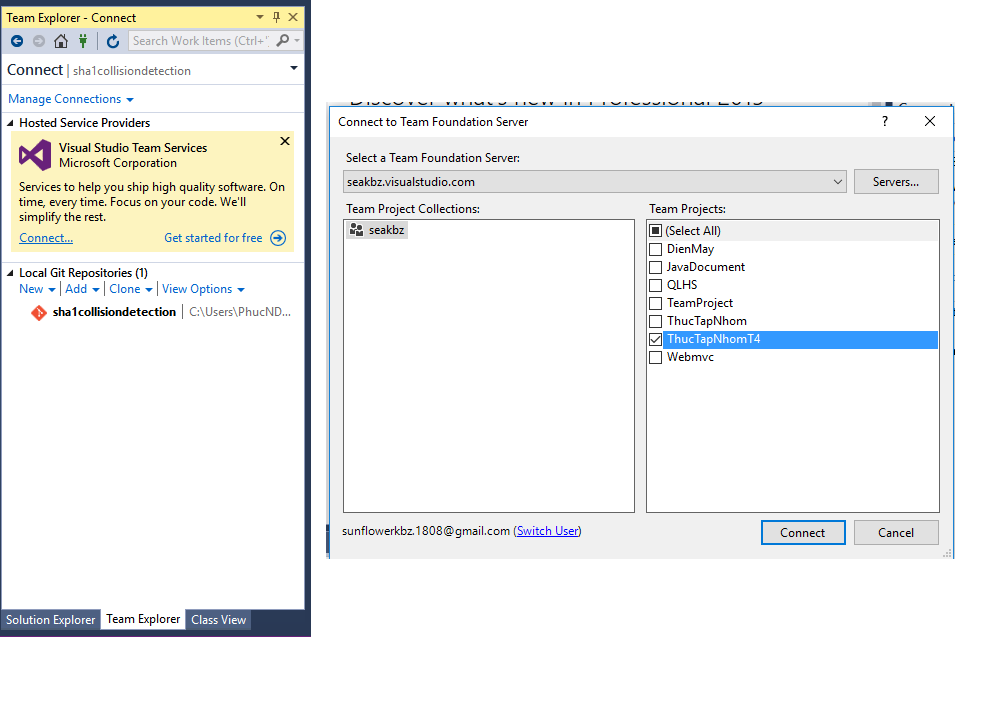


## 2.2 Add Member



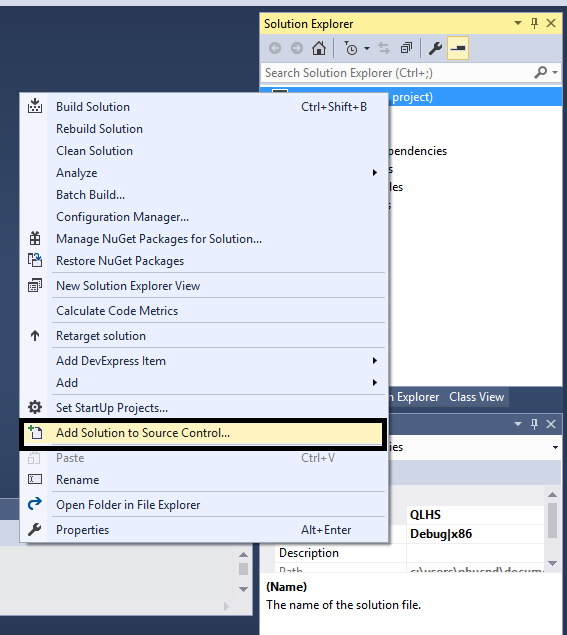
## 2.3 Connect với IDE (visual studio )

* Trong tab **Team Exploer – Connect** Chọn **connect**
* **Select Team Foundation server** (chọn sever làm việc)
* Chọn **Team Project** tích vào dự án tạo ở trên
* Nhấn **connect**

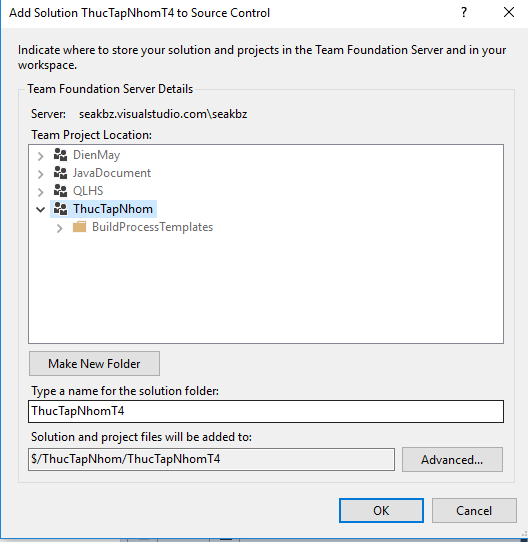


## 2.4 Add Solution to Source Control Up code lên server TFS

* Chuột Phải là **solution** chọn **Add Solution to Source Control**

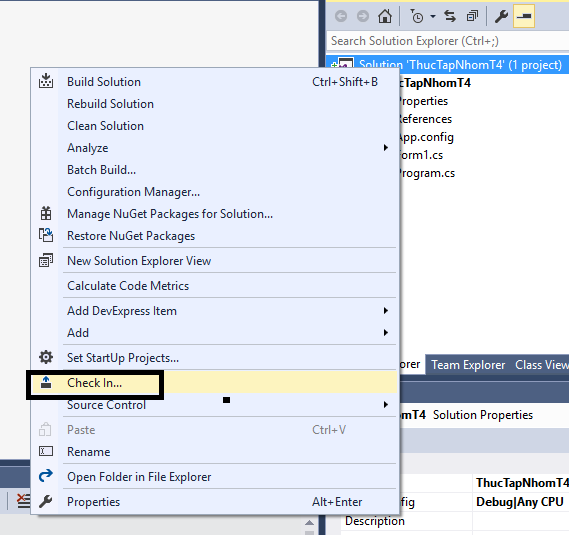


* Chọn vào project mới tạo ở trên

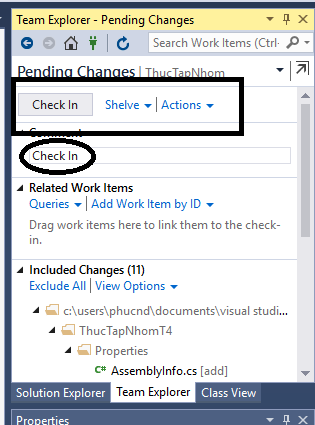


## 2.5 Check in code lên server

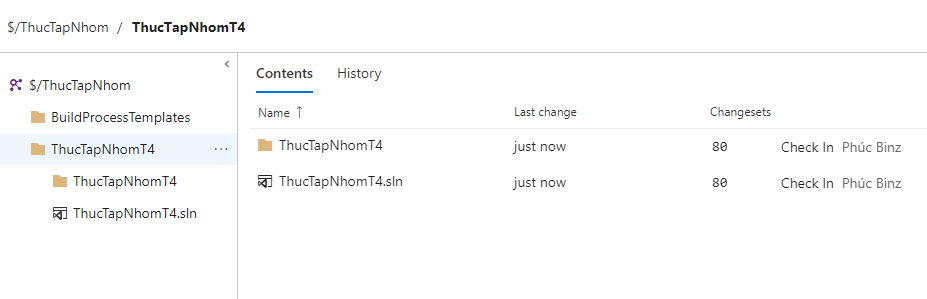
* Chuột phải vào project chọn **check in**



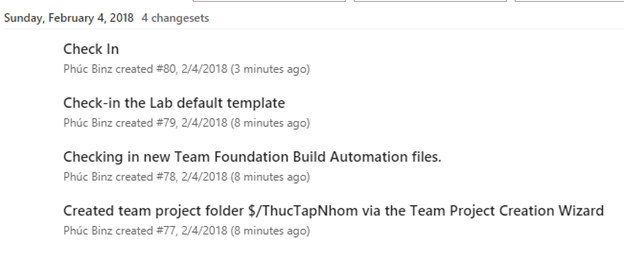
* Màn hình **check in** hiện ra, điền vào chỗ **comment** xem lại các file cần up lên rồi click vào **check in** để đẩy code lên server



* Thay đổi ở server



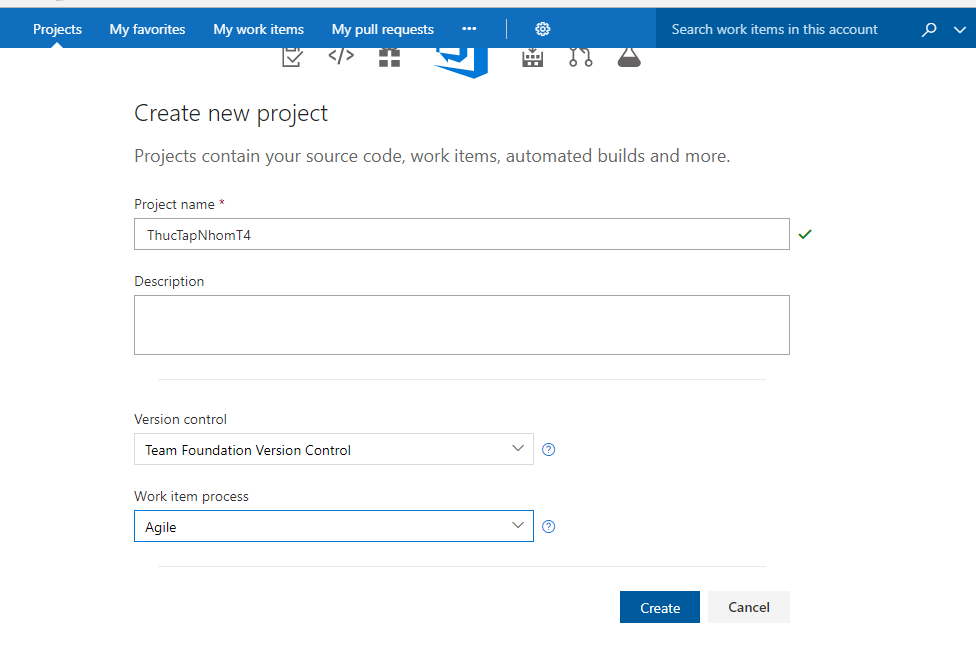
## 2.6 Xem lịch sử chỉnh sửa của thành viên



# 3. Tạo nơi chứa một dự án

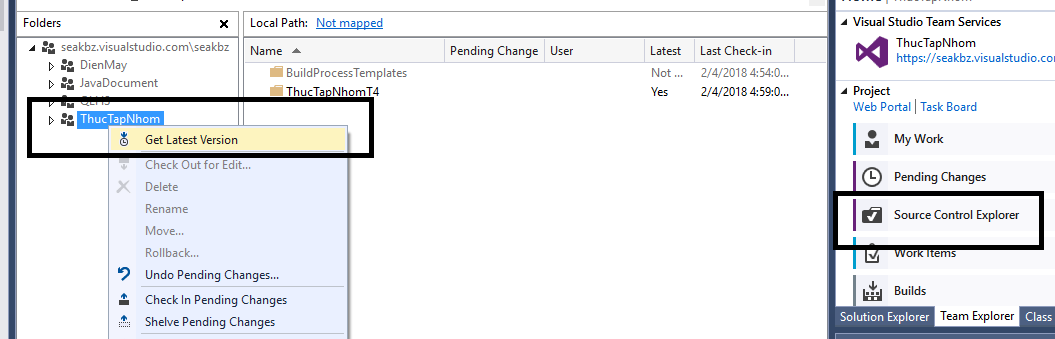
## 3.1 Tạo nơi chứa một dự án ví dụ (thư mục với các files):

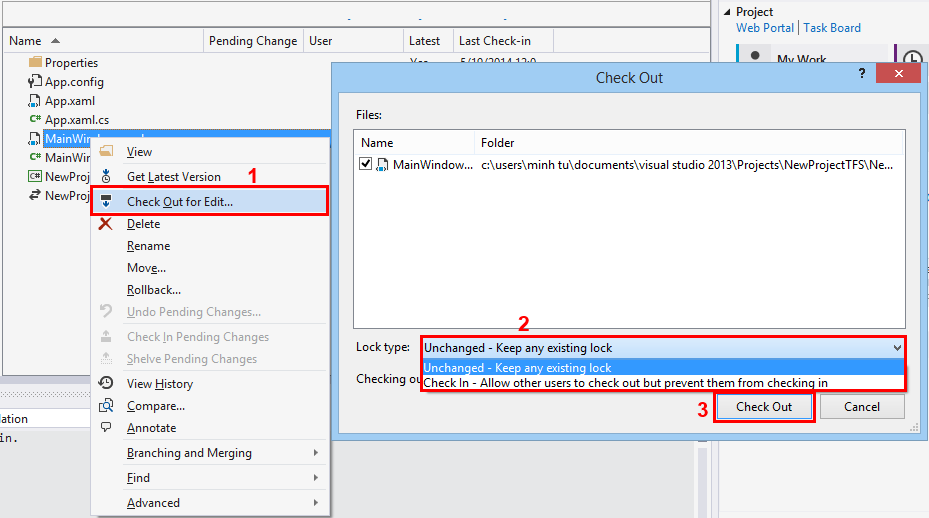
* Vào link : <https://www.visualstudio.com> tạo 1 server
* Sau đó ta có 1 server name có tên <https://seakbz.visualstudio.com> truy cập vào server đó và tạo một project mới



## 3.2 Map code về máy thành viên

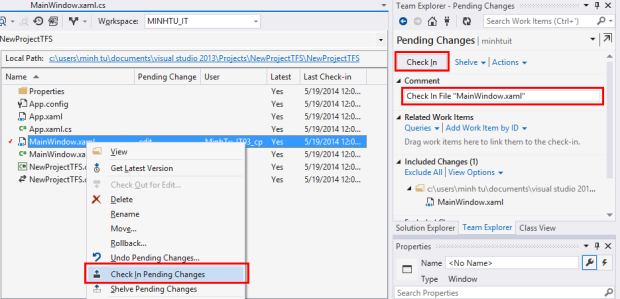
* Ở tab **Team Explorer** Chọn Source **Control Explorer**
* Chuột Phải vào project Chọn **Get Lastest Version**
* Chọn forder map code về và hoàn thành





## 3.3 Check in Code lên server

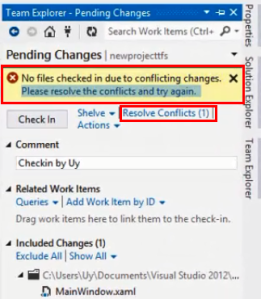
* Sau khi đã thay đổi xong, ta Check In để cập nhật thay đổi lên server.

[](https://minhtuit.files.wordpress.com/2014/05/15.png)

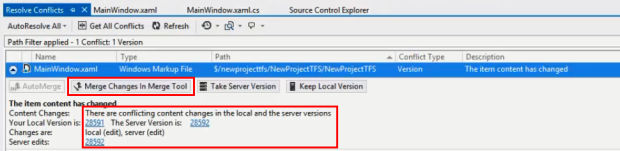
* **Chú ý**: Check Out có 2 sự lựa chọn:
* 1. Unchanged – Keep any existing lock: Cho chép mọi người trong nhóm đều có hể **Check Out**một công việc và khi một thành viên kết thúc trước họ có thể **Check In** để lưu dữ liệu lên hệ thống. Nếu có xung đột xảy ra khi một thành viên khác **Check In** công việc sau thì Visual Studio sẽ thông báo và cho phép thành viên **Check In** sau xử lý lỗi đó (phần này mình sẽ nói ở phía dưới về xử lý xung đột).
* 2. Check In – Allow other users…: Cho chép mọi người trong nhóm đều có thể **Check Out** một công việc nhưng chỉ có thành viên thực hiện việc lựa chọn **Check In** này mới được phép **Check In**để lưu dữ liệu lên hệ thống trước, các thành viên khác phải chờ thành viên này kết thúc việc **Check In** mới có thể **Check In**.

## 3.4 Xử lý xung đột khi một hoặc nhiều người cùng Check In

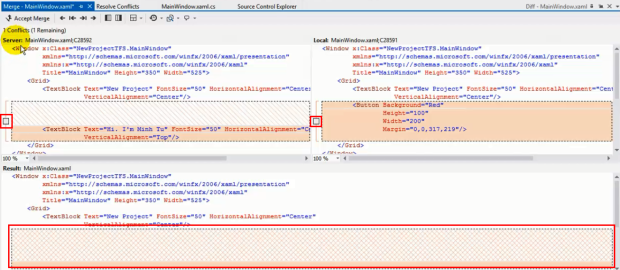
* Việc cùng xin sửa đổi một file và thay đổi file đó sẽ gây xung đột về thay đổi khi 2 người thực hiện sự thay đổi file khác nhau.  
  VD: Cùng một file ban đầu một thanh viên thêm vào một **Button** một thành viên khác đang ở đâu đó và sửa đổi file thêm vào một **TextBlock**. Thành viên thêm vào một **Button** Check In trước thành viên thêm vào một **TextBlock** Check In sau. Visual Studio sẽ tự động thông báo cho người
* Check In sau xung đột và đưa ra thông tin xung đột để giải quyết.

[](https://minhtuit.files.wordpress.com/2014/05/18.png)

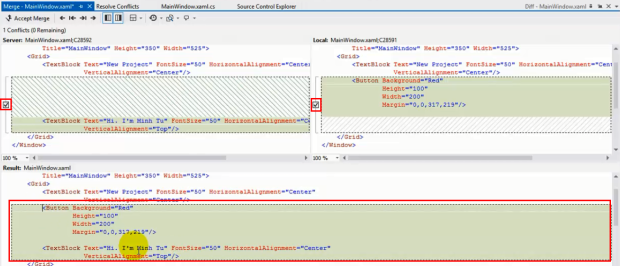
* + Để giải quyết lỗi nhấp chọn **Resolve Conficts**

[](https://minhtuit.files.wordpress.com/2014/05/17.png)

* + Để xem các xung đột nhấp chọn **Merge Changes In Merge Tool** Visual Studio sẽ mở một Tool giúp xử lý các xung đột khi Check I1

[](https://minhtuit.files.wordpress.com/2014/05/19.png)

* + Chỉnh sửa xung đột bằng cách **Check** vào các ô thể hiện các dòng dữ liệu khác nhau.

[](https://minhtuit.files.wordpress.com/2014/05/20.png)

* + Kết thúc việc xử lý xung đột chọn **Accept Merge** và **Check In** lại để hoàn thành công việc.

# 4. Thống kê lưu lượng truy cập của từng thành viên

## 4.1 Thống kê lưu lượng ở giao diện admin

