**Báo cáo Seminar:  
  
Continuous Intergration system :Tìm hiểu các hệ thống quản lý tài liệu, mã nguồn trực tuyến (các loại source control khác nhau), hệ thống tích hợp liên tục**

Các hệ thống chọn lựa:

* + Microsoft visual sourcesafe
  + Git
  + SVN

**Nhóm Project D:**

**Lê Nhiệm – 1459038**

**Thái Hồng Tiên – 1459048**

**Nguyễn Nhật Quang – 1459044**

**Lê Hùng Kiến Luân – 1459032**

Mục lục:

[**Kế hoạch làm việc Seminar** 3](#_Toc489175474)

[**Tự đánh giá kết quả seminar** 4](#_Toc489175475)

[**Đánh giá hoạt động nhóm** 4](#_Toc489175476)

[**Đánh giá thành viên** 5](#_Toc489175477)

[**Thông tin về 3 hệ thống đã thực hiện:** 6](#_Toc489175478)

[Subversion(SVN) 6](#_Toc489175479)

[**Định nghĩa:** 6](#_Toc489175480)

[**Chức năng:** 6](#_Toc489175481)

[**Server-Client** 7](#_Toc489175482)

[Git 8](#_Toc489175483)

[Kho Repo (Repository) 8](#_Toc489175484)

[Remote repository và local repository 9](#_Toc489175485)

[Nhánh (Branch) 9](#_Toc489175486)

[Trộn (Merge) 9](#_Toc489175487)

[Xung đột (Conflict) 9](#_Toc489175488)

[Commit 10](#_Toc489175489)

[Git Remote 10](#_Toc489175490)

[Working Tree và Index 10](#_Toc489175491)

[Kết 10](#_Toc489175492)

[Microsoft Visual SourceSafe (VSS) 11](#_Toc489175493)

[**Chức năng tối thiểu:** 11](#_Toc489175494)

[**Ưu điểm:** 11](#_Toc489175495)

[**Nhược điểm** 11](#_Toc489175496)

[**SO SÁNH** 12](#_Toc489175497)

# **Kế hoạch làm việc Seminar**

**Tuần 1:**

**\_Tìm kiếm thông tin về các hệ thống**

**\_Mỗi người tìm hiểu về cả 3 hệ thống Continuous Intergration system**

**Tuần 2:**

**\_Tìm kiếm thông tin cơ bản và chức năng của từng hệ thống.**

**\_Tìm kiếm các đường link và các trang web liên quan.**

**Tuần 3:**

**\_Tìm hiểu cách sử dụng phần mềm hoặc tool liên quan đến hệ thống**

**Tuần 4:**

**\_Xem xét cách sử dụng cho các thành viên trong nhóm**

**Tuần 5:**

**\_Đánh giá từng hệ thống mà mỗi người trong nhóm chọn**

**\_Chọn ra một hệ thống để demo.**

**Tuần 6:**

**\_So sáng các hệ thống với nhau**

**Tuần 7:**

**\_Tổng hợp thông tin và làm báo cáo.**

# **Tự đánh giá kết quả seminar**

**Hoàn thành 85% của website:**

**\_Tìm hiểu thông tin SVN và hướng dẫn sử dụng, chức năng**

**\_Tìm hiểu thông tin về hệ thông GIT và demo trực tiếp**

**\_Tìm hiểu thông tin về VSS và hướng dẫn sử dụng, chức năng**

**15% chưa hoàn thành:**

**\_Thông tin so sánh còn có sai lệch**

**\_Chưa sử dụng được VSS(tốn phí)**

# **Đánh giá hoạt động nhóm**

* **Hoàn thành dự án chậm hơn kế hoạch 1 tuần.**
* **Có một số vấn đề:** 
  + **1/Thời gian của các thành viên khác nhau: 2 thành viên học AUT, 2 người còn lại không. Do đó có 2 tuần học tập trung AUT làm phân chia công việc thay đổi đột ngột.**
  + **Giải quyết: trong 2 tuần đó giảm công việc cho 2 thành viên AUT, nhưng sau đó tăng lên công việc.**
  + **2/Không thống nhất được thời gian họp nhóm: lý do có đi làm, bận việc nhà, thời gian học khác nhau,..,dẫn đến họp nhóm hằng tuần thay đổi liên tục.**
  + **Vẫn chưa được giải quyết.**
  + **3/Tổng kết so sánh giữa các hệ thống chưa có sự thống nhất.**
  + **4/Một số tool và phần mềm liên quan đến hệ thống có ít thông tin tìm kiếm và khó hiểu.**

# **Đánh giá thành viên**

**Nói chung các thành viên không vi phạm nội quy nhóm đã đặt ra. Nộp thông tin đã tìm hiểu đúng hạn nhưng các thông tin vẫn còn sơ sài và cần bổ sung thêm . Trong các buổi họp nhóm ,các thành viên đến đúng giờ .**

* **Tiên-1459048:Tìm kiếm và nộp thông tin đúng giờ. Có chính kiến về so sánh giữa 2 hệ thống SVN và VSS.**
* **Nhiệm-1459038:Không vi phạm luật nhóm. Hoàn thành phần việc được giao và tộng hợp các thông tin từ thành viện khác trên Facebook, hoặc trên GitHub. Làm presentation.**
* **Nguyễn Nhật Quang-1459044: Hoàn thành công việc được giao đúng hạn, Không vi phạm luật nhóm đã đề ra. Thực hiện demo cho hệ thống Git.**
* **Lê Hùng Kiến Luân-1459032:Hoàn thành phần việc đúng thời hạn. Tìm hiểu rất kĩ về Git và so sánh của nó với SVN.**

# **Thông tin về 3 hệ thống đã thực hiện:**

## Subversion(SVN)

### **Định nghĩa:**

Subversion(viết tắt là SVN) là một hệ thống quản lí version (version control system(VCS)) được giới thiệu vào năm 2000 bởi công ty CollabNet. Đây là hệ thống hỗ trợ làm việc theo nhóm rất hiệu quả. Khi một nhóm làm việc cùng trên một project, việc nhiều người cùng chỉnh sửa nội dung của một file là điều không thể tránh khỏi. SVN cung cấp các chức năng để có thể thực hiện việc này một cách đơn giản và an toàn.

Về mặt khái quát, SVN 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 là nó lưu lại tất cả những gì thay đổi trên hệ thống file: file nào đã bị thay đổi lúc nào, thay đổi

<http://vietcloud.net/thong-tin/10A/0/huong-dan-su-dung-tortoise-svn-subversion-de-quan-ly-code.aspx>

### **Chức năng:**

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.

### **Server-Client**

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.

Phần mềm:  
Cho client: TortoiseSVN, Download: http://tortoisesvn.net/  
Cho server: VisualSVN – Server Download:<http://tortoisesvn.net/downloads.html>

Server:

**Visual SVN** sử dụng như một **Repository Server** (Kho chứa). Dùng để lưu trữ các file dữ liệu chia sẻ giữa các thành viên trong nhóm. **Visual SVN** được cài trên 1 máy chủ, các thành viên trong nhóm sẽ cài đặt chương trình **client**( **subversion client**), có nhiều chương trình**Subversion client**, được nhiều hãng khác nhau phát triển và cơ bản là đều miễn phí.

<http://nghethuatit.com/post/2010/11/08/Huong-dan-cai-c491at-va-su-dung-VisualSVN-Server-va-TortoiseSVN.aspx>

Client:

TortoiseSVN là một trình máy khách, nó quản lý tập tin và thư mục theo thời gian. Các tập tin được lưu trữ trong một kho trung tâm. Kho lưu trữ phần nhiều giống như một máy chủ tập tin bình thường, ngoại trừ việc nó nhớ tất cả các thay đổi đã từng được thực hiện trên các tập tin và thư mục của bạn. Điều này cho phép bạn khôi phục lại phiên bản cũ của tập tin của bạn và kiểm tra lịch sử của việc dữ liệu của bản đã được thay đổi như thế nào và khi nào, và ai đã thay đổi nó.

<https://tortoisesvn.net/docs/release/TortoiseSVN_vi/tsvn-preface.html>

## Git

Git là một trong những Hệ thống Quản lý Phiên bản Phân tán, vốn được phát triển nhằm quản lý mã nguồn (source code) của Linux.  
  
Trên Git, ta có thể lưu trạng thái của file dưới dạng lịch sử cập nhật. Vì thế, có thể đưa file đã chỉnh sửa một lần về trạng thái cũ hay có thể biết được file đã được chỉnh sửa chỗ nào.  
  
Thêm nữa, khi định ghi đè (overwrite) lên file mới nhất đã chỉnh sửa của người khác bằng file đã chỉnh sửa dựa trên file cũ, thì khi upload lên server sẽ hiện ra cảnh cáo. Vì thế, sẽ không xảy ra lỗi khi ghi đè lên nội dung chỉnh sửa của người khác mà không hề hay biết.  
  
Git sử dụng mô hình phân tán, ngược lại so với SVN hoặc CSV. Mỗi nơi lưu source sẽ đc gọi là repositories, không cần lưu trữ tập trung một nơi, mà mỗi thành viên trong team sẽ có một repository ở máy của riêng mình.  
Điều đó có nghĩa là nếu có 3 người A,B,C cùng làm việc trong 1 project. Thì bản thân repo trên máy của người A, người B, và người C có thể kết nối được với nhau.  
  
Khi quyết định thay đổi chỗ nào đó lên server ta chỉ cần một thao tác "push" nó lên server. Chúng ta vẫn có thể share thay đổi của chúng ta cho thành viên khác, bằng cách commit hoặc update trực tiếp từ máy của họ mà không phải thông qua repositories gốc trên server (thông qua share ssh cho nhau).  
  
Lợi ích

* An toàn hơn (vì mỗi bản copy của thành viên đều là full copy từ repository gốc, khi server bị down).
* Các thành viên vẫn có thể làm việc offline, họ vẫn có thể commit và update trên local của họ hoặc thậm chí với nhau mà không cần thông qua server.
* Khi server hoạt động trở lại, họ có thể cập nhật tất cả lên lại server.

Tiếp đây mình xin giới thiệu về các khái niệm mà bạn cần nắm trong git:

### Kho Repo (Repository)

Repository hay được gọi tắt là Repo, đơn giản là nơi chứa tất cả những thông tin cần thiết để duy trì và quản lý các sửa đổi và lịch sử của toàn bộ project. Trong Repo có 2 cấu trúc dữ liệu chính là Object Store và Index. Tất cả dữ liệu của Repo đèu được chứa trong thư mục bạn đang làm việc dưới dạng folder ẩn có tên là .git

### Remote repository và local repository

Đầu tiên, repository của Git được phân thành 2 loại là remote repository và local repository.

* Remote repository: Là repository để chia sẻ giữa nhiều người và bố trí trên server chuyên dụng.
* Local repository: Là repository bố trí trên máy của bản thân mình, dành cho một người dùng sử dụng.

Do repository phân thành 2 loại là local và remote nên với những công việc bình thường thì có thể sử dụng local repository. Khi muốn public nội dung công việc mà mình đã làm trên local repository, thì ta sẽ upload lên remote repository rồi public. Thêm nữa, thông qua remote repository bạn cũng có thể lấy về nội dung thay đổi của người khác.

Nguồn : www.git-tower.com

### Nhánh (Branch)

Đây là một trong những thế mạnh của git là nhánh. Với git, việc quản lý nhánh rất dễ dàng. Mỗi nhánh trong Git gần giống như một workspace. Việc nhảy vào một nhánh để làm việc trong đó tương tự việc chuyển qua ngữ cảnh làm việc mới, và sau đó có thể nhanh chóng quay lại ngữ cảnh cũ.  
  
Nhánh (branch) được dùng để phát triển tính năng mới mà không làm ảnh hưởng đến code hiện tại.  
  
Nhánh master là nhánh “mặc định” khi bạn tạo một repository.  
Nhánh master thông thường là nhánh chính của ứng dụng. Ví dụ bạn thử nghiệm một tính năng mới và muốn không ảnh hưởng đến code chính bạn có thể tạo một nhánh mới và sau khi xong sẽ hợp nhất lại với nhánh master. Việc hợp nhất 2 nhánh lại được gọi là merge.

Nguồn : <http://rogerdudler.github.io/git-guide/index.vi.html>

### Trộn (Merge)

Trộn source từ một nhánh khác vào nhánh hiện tại. Bạn sẽ hiểu sâu hơn về nó trong các ví dụ ở các bài sau.  
Chú ý:

* Kiểm tra branch hiện đang làm việc trước khi merge
* Phải đẩy tất cả những thay đổi dưới máy local lên Git trước khi merge
* Trước khi merge phải lấy hết những thay đổi mới nhất của các branch khác, hay ít nhất là branch cần merge về máy
* Merge thành công thì nên đẩy source lên lại lên server
* Nên merge bằng GUI tool.

### Xung đột (Conflict)

Conflic là trường hợp có 2 sự thay đổi trong một dòng code và máy tính không thể tự quyết định dòng code nào là “đúng”. Đúng ở đây có nghĩa là “ý đồ của lập trình viên”.

Để giải quyết mâu thuẫn bạn phải dùng “tay không” để sữa các xung đột này. Bạn chỉ việc nhìn vào file bị conflict và tự quyết định dòng code nào giữ lại, dòng nào xóa bỏ.

### Commit

Để ghi lại việc thêm/thay đổi file hay thư mục vào repository thì sẽ thực hiện thao tác gọi là Commit.

Khi thực hiện commit, trong repository sẽ tạo ra commit (hoặc revision) đã ghi lại sự khác biệt từ trạng thái đã commit lần trước với trạng thái hiện tại.

Commit này đang được chứa tại repository, các commit nối tiếp với nhau theo thứ tự thời gian. Bằng việc lần theo commit này từ trạng thái mới nhất thì có thể biết được lịch sử thay đổi trong quá khứ hoặc nội dung thay đổi đó.

Các commit này, được đặt tên bởi 40 ký tự alphabet (mã md5 thì phải) không trùng nhau được băm từ thông tin commit. Bằng việc đặt tên cho commit, có thể chỉ định commit nào từ trong repository.

Mỗi commit đều có yêu cầu phải có commit message, để giải thích commit này là bạn đã làm gì trong này.

### Git Remote

Để kết nối được với một repo khác người ta sử dụng một khái niệm gọi là remote.

Trên thực tế khi làm việc với nhau thì không như vậy, vì không phải máy ai cũng cài một “git server” để người khác kết nối được với mình. Thông thường thì chúng ta sẽ sử dụng một repo chung và các máy kết nối vào repo đó.

Có 2 “git repo server” được sử dụng nhiều là github.com và bitbucket.org.

Trên thực tế khi có 2 người cùng làm việc với 1 project thì thông thường sẽ tạo một repo trên github hoặc bitbucket và repo trên máy người A sẽ kết nối với repo trên github và máy người B cũng kết nối với repo trên github/bitbucket. Từ đó source code của người A và người B sẽ được đồng bộ với nhau thông qua repo trên github/bitbucket.

Vì vậy, trước khi sử dụng git thì bạn nên đăng kí một tài khoản trên github.com hoặc bitbucket.org.

### Working Tree và Index

Trên Git, những thư mục được đặt trong sự quản lý của Git mà mọi người đang thực hiện công việc trong thực tế được gọi là working tree.

Và trên Git, giữa repository và working tree tồn tại một nơi gọi là index. Index là nơi để chuẩn bị cho việc commit lên repository.

Nguồn : backlogtool

Trên Git, khi đã thực hiện commit thì trạng thái sẽ không được ghi trực tiếp trong repository từ working tree, mà sẽ ghi trạng thái đã được thiết lập của index được xây dựng ở giữa đó. Vì thế, để ghi lại trạng thái của file bằng commit thì trước hết cần thông báo file trong index.

### Kết

Còn khá nhiều định nghĩa trên Git, trên đây mình chỉ nói một vài định nghĩa quan trọng, mình sẽ cố gắng cập nhật đầy đủ sau.

Ở bài sau mình sẽ hướng dẫn cách sử dụng Git, sử dụng github, các làm việc trên git và xử lý các trường hợp thường gặp khi teamwork với Git.

## Microsoft Visual SourceSafe (VSS)

**Định nghĩa:**

Là một chương trình kiểm soát nguồn không tiếp tục, hướng đến các dự án phát triển phần mềm nhỏ.

Tổng quan Source Safe nguyên thủy được sản xuất bởi 1 công ty tên là “One Tree Software” . Version đầu tiên được phát hành là bản 3.1 , đó là 1 chương trình 16 bit. Microsoft thời điểm đó cũng có 1 hệ thống quản lý source code yếu hơn là Delta . Năm 1994 , Microsoft mua lại công ty “One Tree Software” và cập nhật lại phiên bản 3.1 . Kết quả là sự ra đời phiên bản 4.0 của visual Source Safe . Đó là phiên bản 32 bit . Năm 1995 , nó được phát hành rộng rãi

Như hầu hết các hệ thống quản lý cấu hình khác, Source Safe tạo 1 thư viện ảo trên các tập tin máy tính . Người dùng có thể đọc bất kì file nào tại bất kì thời điểm nào ,nhưng để thay đổi nó , người dùng phải “check out” nó trước . Sau khi chúng được sửa chữa , phải “check in” nó lại . Những thay đổi sẽ được cập nhật đối với tất cả người dùng sau khi nó được “check in”

Visual sourcesafe hỗ trợ phát triển cho nhiều nền tảng bằng cách cho phép chia sẻ và chỉnh sửa dữ liệu.

Ví dụ như 1 cơ sở mã nguồn phần mềm trên nhiều hệ điều hành, visual sourcesafe giúp developers sử dụng lại hoặc object oriented cái code và giúp theo dõi các ứng dụng sử dụng các số liệu mô đun cụ thể.

### **Chức năng tối thiểu:**

\_Giúp bảo vệ nhóm khỏi sự cố mất file

\_cho phép truy ngược lại phiên bản trước đó của file

\_Hỗ trợ chia nhánh, chia sẻ, hợp nhất và quản lý việc phát hành file

\_theo dõi phiên bản của toàn dự án

\_Theo dõi mã module (1 file được sử dụng lại hoặc chia sẻ bởi nhiều dự án)

### **Ưu điểm:**

\_Cho phép chia sẻ nhanh chóng và hiệu quả các tệp giữa các dự án.

\_khi thêm file vào visual sourcesafe, file này được lưu trên cơ sở dữ liệu và tạo sẵn cho người dùng khác.

\_có thể khổi phục phiên bản cũ bất kì lúc nào

### **Nhược điểm**

Tính ổn định của Visual SourceSafe bị chỉ trích do cách Visual SourceSafe sử dụng cơ chế truy cập trực tiếp, dựa trên tệp tin, cho phép bất kỳ máy khách nào sửa đổi một tệp tin trong kho sau khi khóa nó. Nếu một máy khách gặp sự cố giữa bản cập nhật một tập tin, nó có thể hỏng tập tin đó. Nhiều người dùng Visual SourceSafe làm giảm nguy cơ này bằng cách sử dụng một tiện ích được cung cấp bởi Visual SourceSafe để kiểm tra cơ sở dữ liệu để tham nhũng và, khi có thể, sửa lỗi mà nó tìm thấy.

### **SO SÁNH**

Visual source safe:  
- Được tích hợp sẵn vào Visual studio nên xài rất thuận tiện - Sử dụng 2 cơ chế là check in 1 người và nhiều người ( nếu 1 người check in thì người khác ko thể checkin, do đó sẽ không bị vấn đề conflic)  
  
SVN:  
- tích hợp vào [Windows[http://cdncache-a.akamaihd.net/items/it/img/arrow-10x10.png](http://s.igmhb.com/click?v=Vk46MTM1NTQ2OjY0MzY6d2luZG93OjMwODQ4ZmJkZmRkOGI0NjA4MGUyZmI1M2JiNDM2M2NhOnotMjIwMi04NzczMzg1OTp3d3cuZGR0aC5jb206Mzk0OTEyOjMwZTlkMWIyMTBhYjdkYWU3ODg5ZWVlOTk0OTMxNjM1OmU4MTY5OWEyYTFlNTQ0ZWZiNDc1OThkYTUxODFlMDdmOjE6ZGF0YV9zcywxMDQweDI1NjA7ZGF0YV9yYywyO2RhdGFfZmIsbm87OjQ0MDA3NTA6OjowLjAx&subid=g-87733859-3f408e82e9584aecba3536d48aedbc97-&data_ss=1040x2560&data_rc=2&data_fb=no&data_tagname=A&data_ct=image_only&data_clickel=link&data_sid=c1b469311b9aaeaf2bf90886cda665a7)](http://s.igmhb.com/click?v=Vk46MTM1NTQ2OjY0MzY6d2luZG93OjMwODQ4ZmJkZmRkOGI0NjA4MGUyZmI1M2JiNDM2M2NhOnotMjIwMi04NzczMzg1OTp3d3cuZGR0aC5jb206Mzk0OTEyOjMwZTlkMWIyMTBhYjdkYWU3ODg5ZWVlOTk0OTMxNjM1OmU4MTY5OWEyYTFlNTQ0ZWZiNDc1OThkYTUxODFlMDdmOjE6ZGF0YV9zcywxMDQweDI1NjA7ZGF0YV9yYywyO2RhdGFfZmIsbm87OjQ0MDA3NTA6OjowLjAx&subid=g-87733859-3f408e82e9584aecba3536d48aedbc97-&data_ss=1040x2560&data_rc=2&data_fb=no&data_tagname=A&data_ct=image_only&data_clickel=link&data_sid=c1b469311b9aaeaf2bf90886cda665a7) explorer, mỗi khi cập nhật phải vào đúng thư mục rồi cập nhật, rất bất tiện  
( nếu muốn tích hợp vào trong Visual studio thì bạn phải cài đặt thêm Visual SVN, cái này phải tốn tiền mua)  
- rất dễ bị conflic nếu sửa file mà không update trước, nếu đã bị conflic thì bạn phải mở chức năng so sánh 2 phiên bản rồi tiến hành sửa