

CHƯƠNG 2

PHÁT TRIỂN PHẦN MỀM NGUỒN MỞ

Nội dung giảng dạy

- Cộng đồng phát triển phần mềm nguồn mở
- Tiến trình phát triển phần mềm nguồn mở
- Hệ thống quản lý phiên bản mã nguồn



Cộng đồng phát triển phần mềm nguồn mở

- Cộng đồng là một nhóm cá nhân chia sẻ một/một vài sự quan tâm chung nào đó
 - Cả phần mềm nguồn đóng và mở đều có các cộng đồng của chúng
 - Phần lớn các thành viên tham gia để trao đổi, tương tác với các thành viên khác trong cộng đồng
 - Một dạng cộng đồng khác trong đó một số thành viên có các vai trò đặc biệt hơn, có quyền quyết định các vai trò hoạt động khác, ví dụ, báo cáo lỗi, viết tài liệu, hỗ trợ các thành viên khác,...

Phát triển phần mềm nguồn mở

- Nhu cầu chia sẻ mã nguồn giữa:
 - Nhóm phát triển
 - Trong cùng một công ty
 - Cộng đồng cùng phát triển
- Ý tưởng của Stallman (đang làm việc tại Phòng thí nghiệm Trí tuệ Nhân tạo của MIT) chia sẻ mã nguồn cho một cộng đồng rộng rãi cùng phát triển bắt đầu từ 1970
- Phát triển phần mềm nguồn mở là một tiến trình phát triển phần mềm trong đó mã nguồn được công bố



Cộng đồng phát triển phần mềm nguồn mở

- Lợi thế của cộng đồng nguồn mở:
 - Nhiều bình duyệt sẽ làm cho chất lượng, tính an toàn
 - Có thể sao chép nhanh chóng mà không lo lắng về chi phí (miễn phí hoặc thấp)
 - Có thể chia sẻ chi phí phát triển với những người trong cộng đồng
 - Có thể sửa đổi trong các trường hợp cần thiết

Phát triển phần mềm nguồn mở

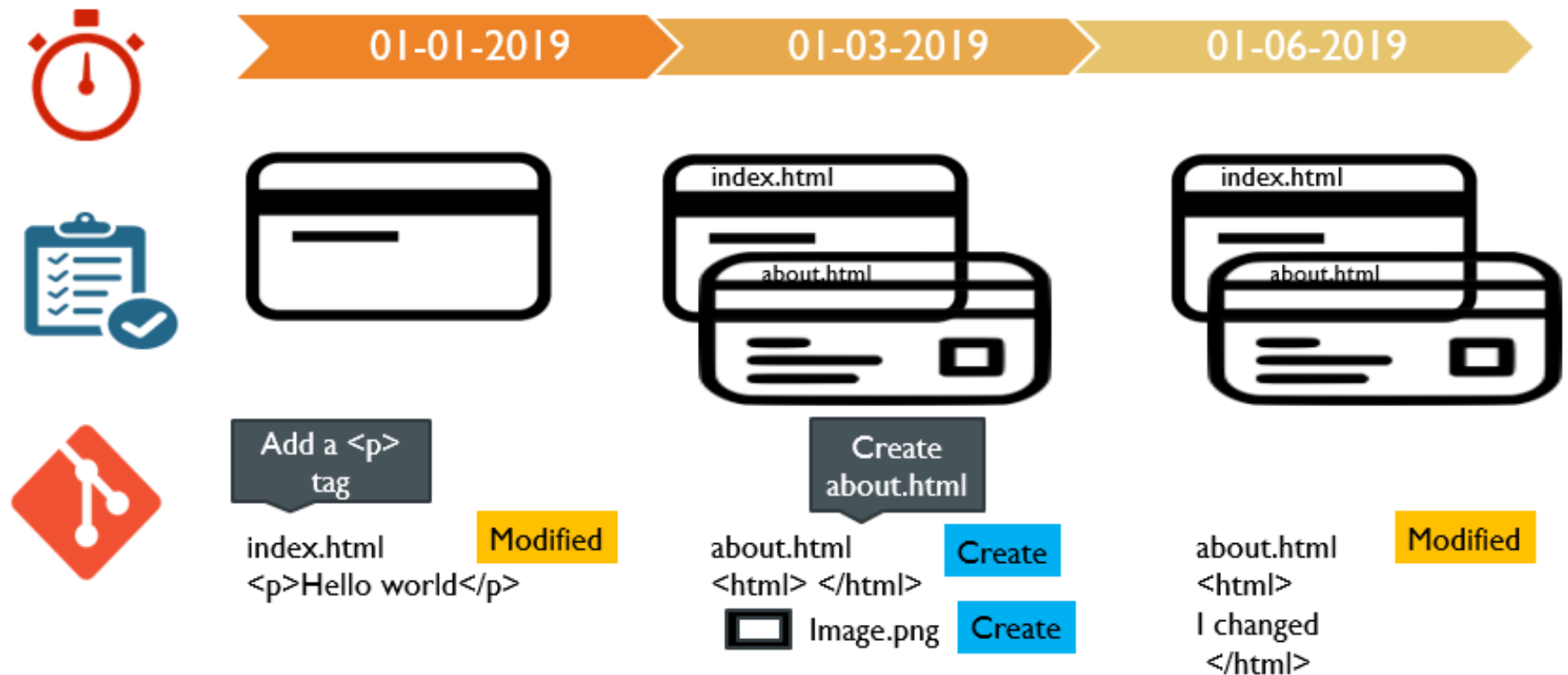
- Các hoạt động trong phát triển phần mềm:
 - Đặc tả: định nghĩa việc hệ thống làm gì
 - Thiết kế: định nghĩa việc hệ thống được tổ chức và thực thi như thế nào
 - Xây dựng: tiến hành việc xây dựng hệ thống
 - Kiểm chứng (kiểm thử): kiểm tra việc hệ thống đã thực hiện đúng những gì mà khách hàng muốn hay chưa
 - Triển khai: đưa hệ thống vào sử dụng, vận hành
 - Tiến hoá (cập nhật): thay đổi hệ thống để đáp ứng các thay đổi như khách hàng cần

Phát triển phần mềm nguồn mở

- Một số vấn đề liên quan khi phát triển phần mềm nguồn mở:
 - Người tham gia phát triển khắp nơi trên thế giới
 - Tham gia cải tiến mã nguồn cùng lúc
 - Luôn có nhu cầu chỉnh sửa mã nguồn trên phiên bản mới nhất
- Cần phải giải quyết những vấn đề sau:
 - Quy trình, phương thức quản trị việc phát triển
 - Mức độ mở
 - Lựa chọn công cụ hỗ trợ việc phát triển sản phẩm một cách đồng thời

Phát triển phần mềm nguồn mở

- Một số vấn đề liên quan khi phát triển phần mềm nguồn mở:





Hệ thống quản lý phiên bản mã nguồn

- VCS (Version Control Systems)
 - Là hệ thống lưu giữ các phiên bản của mã nguồn của sản phẩm phần mềm theo thời gian
 - VCS cho phép người quản trị phân chia các tập tin cho từng thành viên, VD: lập trình viên quản lý file code, người thiết kế đồ họa quản lý các file thiết kế,...
 - Các chức năng cơ bản của VCS:
 - Xem lại những thay đổi của file theo thời gian
 - Xem ai là người đã thực hiện các thay đổi
 - Khôi phục lại trạng thái của file trước khi thay đổi ở một thời điểm nào đó



Hệ thống quản lý phiên bản mã nguồn

- VCS (Version Control Systems)
 - Cho phép các thành viên có thể phát hiện lỗi và sửa lỗi thuận tiện
 - Tự động so sánh và cập nhật mã nguồn khi các thành viên sửa đổi cùng một tập tin tại cùng một thời điểm
 - ...

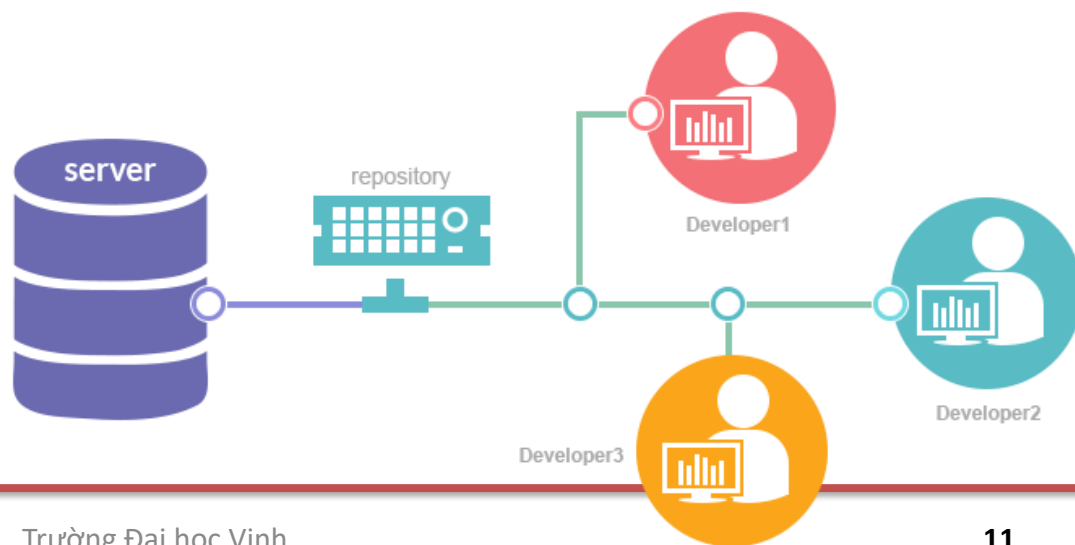


TOP VERSION CONTROL SYSTEMS



Hệ thống quản lý phiên bản mã nguồn

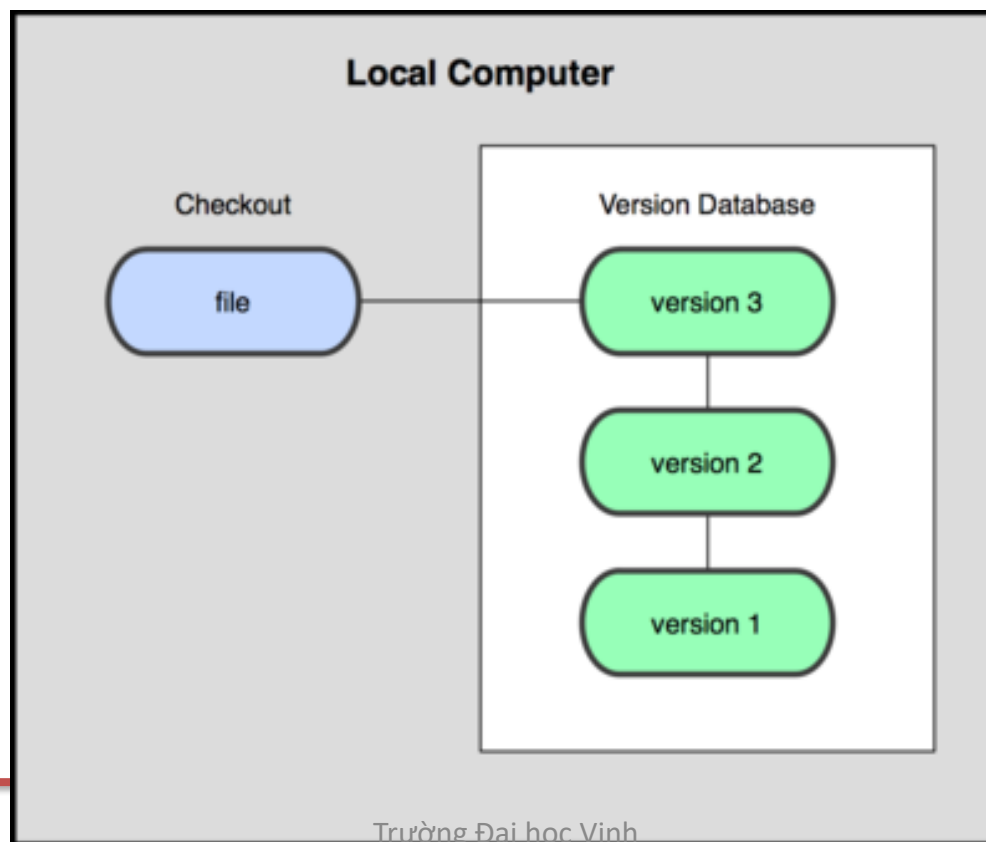
- VCS (Version Control Systems): có 3 loại hệ thống quản lý phiên bản:
 - Quản lý cục bộ (Local Version Control System - LVCS)
 - Quản lý tập trung (Centralized Version Control System – CVCS)
 - Quản lý phân tán (Distributed Version Control System – DVCS)





Hệ thống quản lý phiên bản mã nguồn

- Quản lý cục bộ (Local Version Control System)
 - Máy tính cá nhân sẽ lưu trữ tất cả các thay đổi của file
 - Một người làm việc và chỉ biết phiên bản làm việc của mình





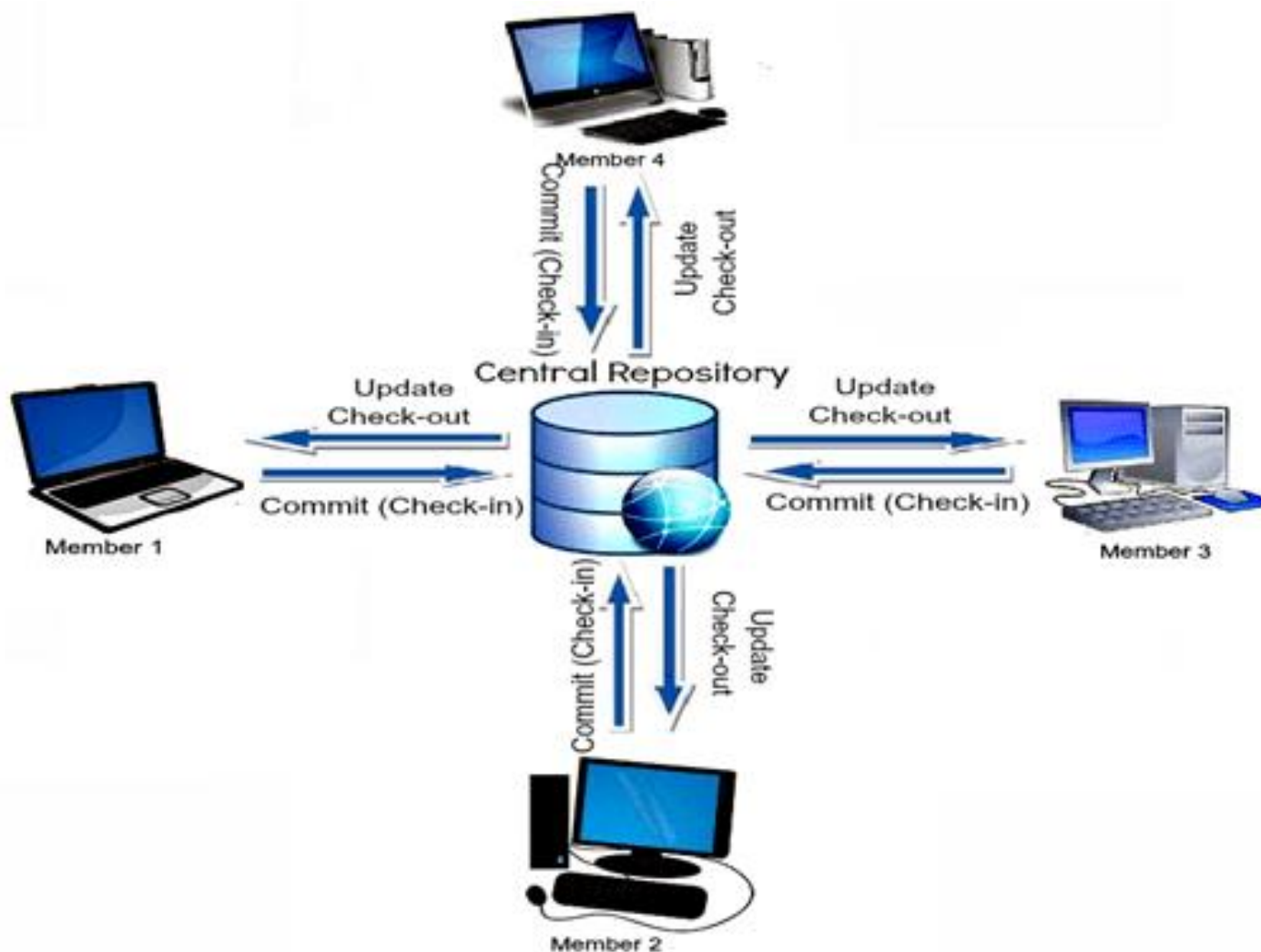
Hệ thống quản lý phiên bản mã nguồn

- Quản lý tập trung (Centralized Version Control System – CVCS):
 - Tất cả các folder, file của dự án sẽ nằm tập trung tại 1 server lớn duy nhất (Central Repository)
 - Tất cả các thành viên tham gia dự án có thể lấy code về (clone), chỉnh sửa code, sau đó commit trở lại central repository
 - Tất cả các chức năng của 1 version control (theo dõi file, gộp code...) đều nằm ở central server, các thành viên tham gia chỉ đơn giản là sử dụng 1 chương trình client kết nối với central server thông qua network



Hệ thống quản lý phiên bản mã nguồn

- Quản lý tập trung (Centralized Version Control System – CVCS):





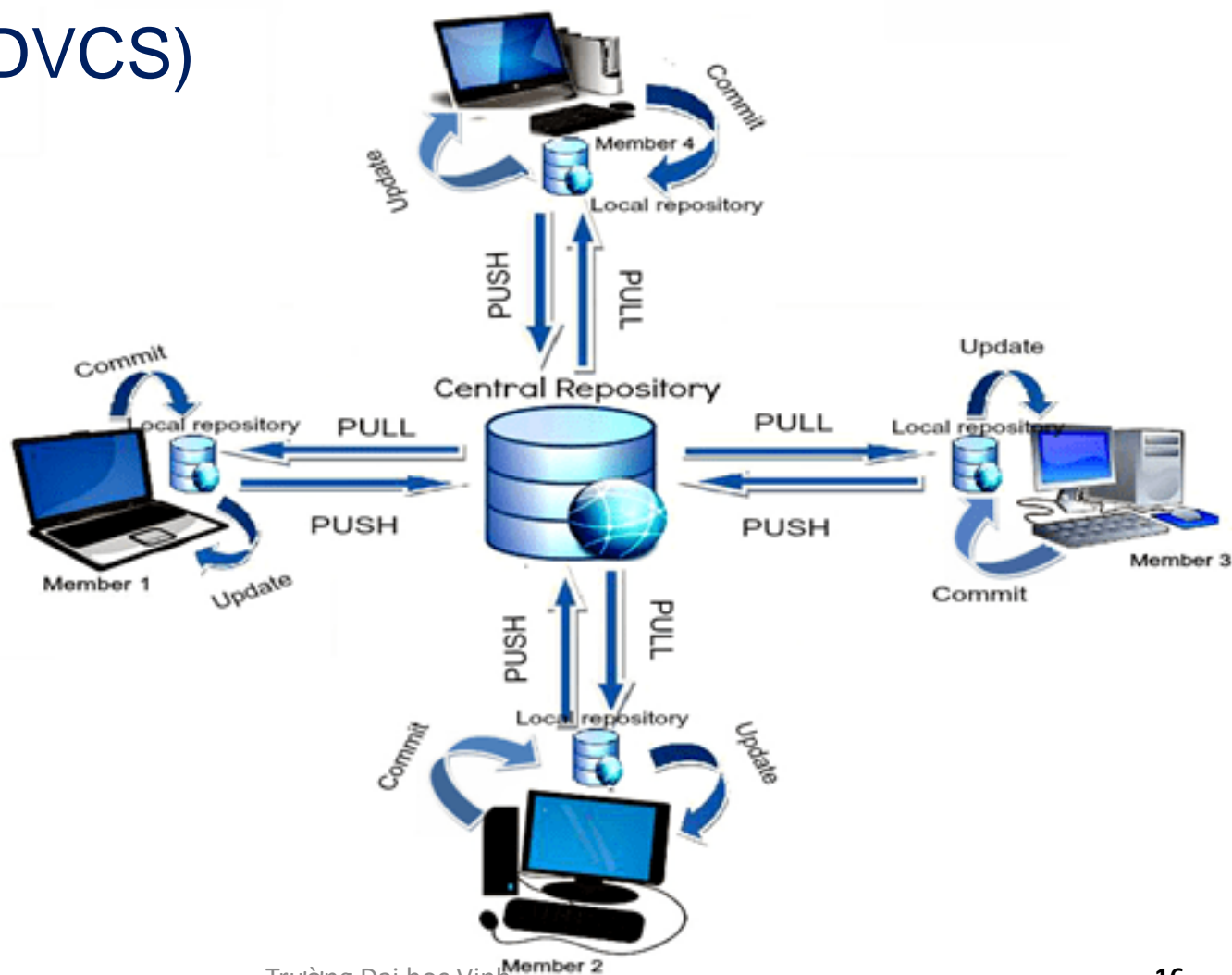
Hệ thống quản lý phiên bản mã nguồn

- Quản lý phân tán (Distributed Version Control System – DVCS)
 - Mỗi thành viên tham gia dự án sẽ có 1 local repository tại máy cá nhân của mình
 - Mỗi người đều có thể tạo ra các branch, commit code, gộp code tại máy riêng của mình mà không ảnh hưởng gì đến central repository, không cần đến network
 - Sau đó, các thành viên có thể gộp local repository và central repository lại thành một thông qua phương thức push
 - Ưu điểm: các máy khách có thể làm việc độc lập khi không kết nối được với server



Hệ thống quản lý phiên bản mã nguồn

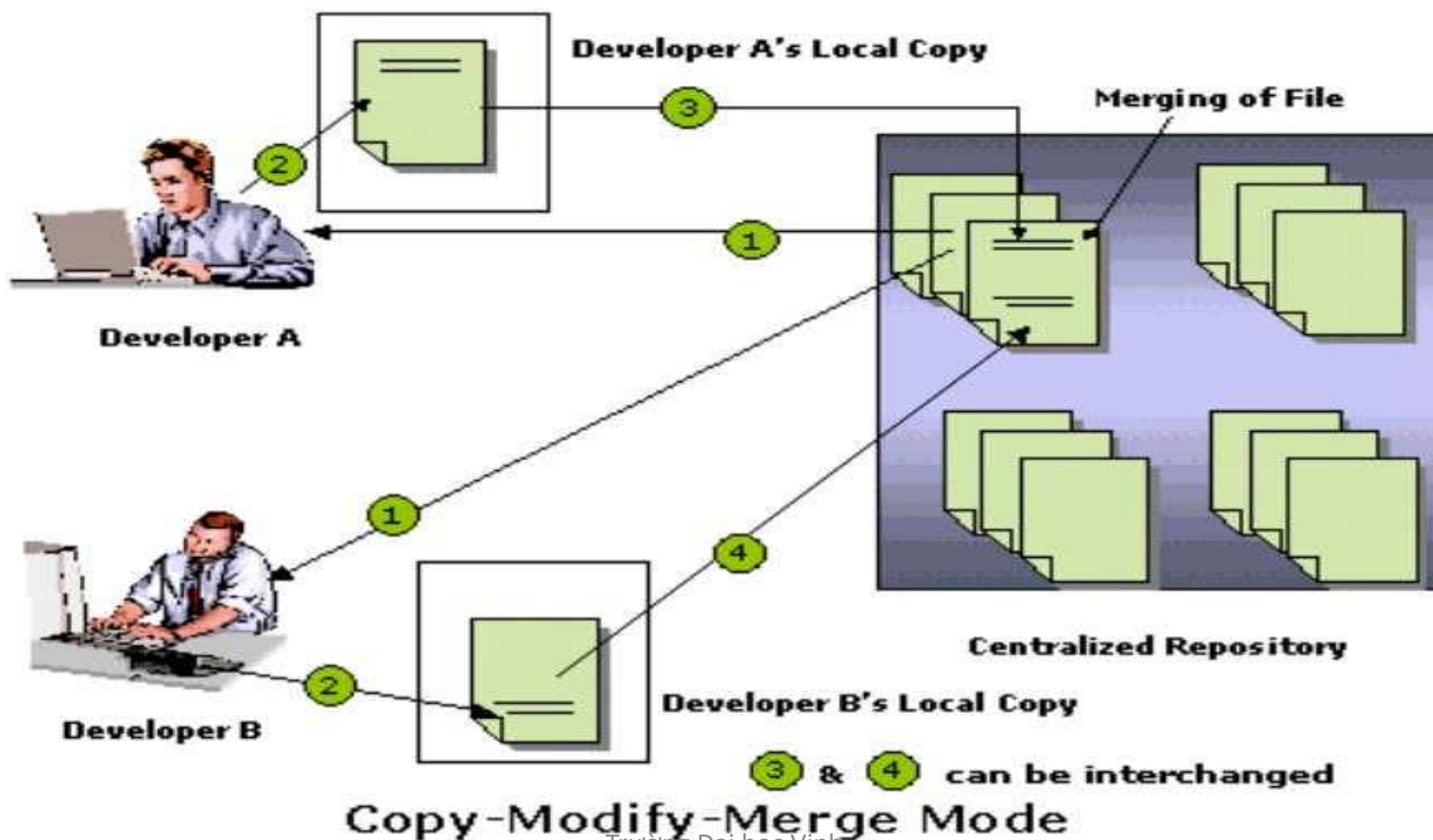
- Quản lý phân tán (Distributed Version Control System – DVCS)





Hệ thống quản lý phiên bản mã nguồn

- Mô hình hoạt động tổng quan:
 - Sao chép – Sửa đổi – Gộp





Hệ thống quản lý phiên bản mã nguồn

- Quy trình làm việc:
 - Người phát triển (developer) tải một bản copy đang làm việc của chương trình từ hệ thống (check out)
 - Người phát triển chỉnh sửa bản sao chép một cách tự do. Cùng thời điểm, những người phát triển khác cũng đang làm việc trên bản copy của họ
 - Người phát triển kết thúc việc thay đổi và chuyển nó cho hệ thống quản lý mã nguồn (commit)
 - Các nhà phát triển khác có thể yêu cầu truy vấn thư viện các bản copy để thấy bản copy chính (master copy) có thay đổi hay không. Nếu có, hệ thống tự động cập nhật các bản copy đang làm việc của họ



Hệ thống quản lý phiên bản mã nguồn

- Một số hệ thống quản lý phiên bản mã nguồn
 - CVS (Concurrent Versions System)
 - Subversion (SVN)
 - Git
 - Github
 - SourceForge
 - Mercurial
 - ...





Hệ thống quản lý phiên bản mã nguồn

- CVS: Hệ thống các phiên bản đồng thời
 - Đây là hệ thống quản lý phần mềm mã nguồn mở, được phát triển vào những năm 80
 - CVS được dùng để theo vết tất cả các công việc và các thay đổi trong một tập hợp các tập tin
 - CVS cho phép nhiều lập trình viên phát triển phần mềm cộng tác với nhau
 - CVS được cấp phép bởi giấy phép công cộng GNU GPL





Hệ thống quản lý phiên bản mã nguồn

- CVS: Một số đặc điểm
 - CVS sử dụng kiến trúc Client-Server
 - Server lưu trữ các phiên bản hiện tại của dự án và lịch sử của nó
 - Các client kết nối tới server để lấy về một phiên bản copy đầy đủ dự án, làm việc trên phiên bản đó và sau đó lưu lại nhưng thay đổi của họ
 - Thông thường, client và server kết nối thông qua mạng LAN hoặc Internet, Client và Server có thể cùng chạy trên một máy nếu CVS có nhiệm vụ theo vết lịch sử của dự án do các nhà phát triển phần mềm phát triển trong nội bộ



Hệ thống quản lý phiên bản mã nguồn



VISUALSVN

- Subversion (SVN):
 - Là phần mềm mã nguồn mở dùng để quản lý và kiểm tra các phiên bản mã nguồn khác nhau trong quá trình phát triển phần mềm
 - Subversion được thiết kế với mục đích thay thế phần mềm quản lý phiên bản Concurrent Versioning System (CVS) đã cũ và có nhiều nhược điểm
 - Một số phần mềm hay dùng: Visual SVN Server (server), Tortoise SVN (client) hoặc website XP-Dev
 - Subversion rất phổ biến trong giới mã nguồn mở và được dùng để quản lý mã nguồn cho nhiều ứng dụng phần mềm miễn phí như: Apache Software Foundation, KDE, Free Pascal, FreeBSD, GCC, Python, Ruby,...



TortoiseSVN



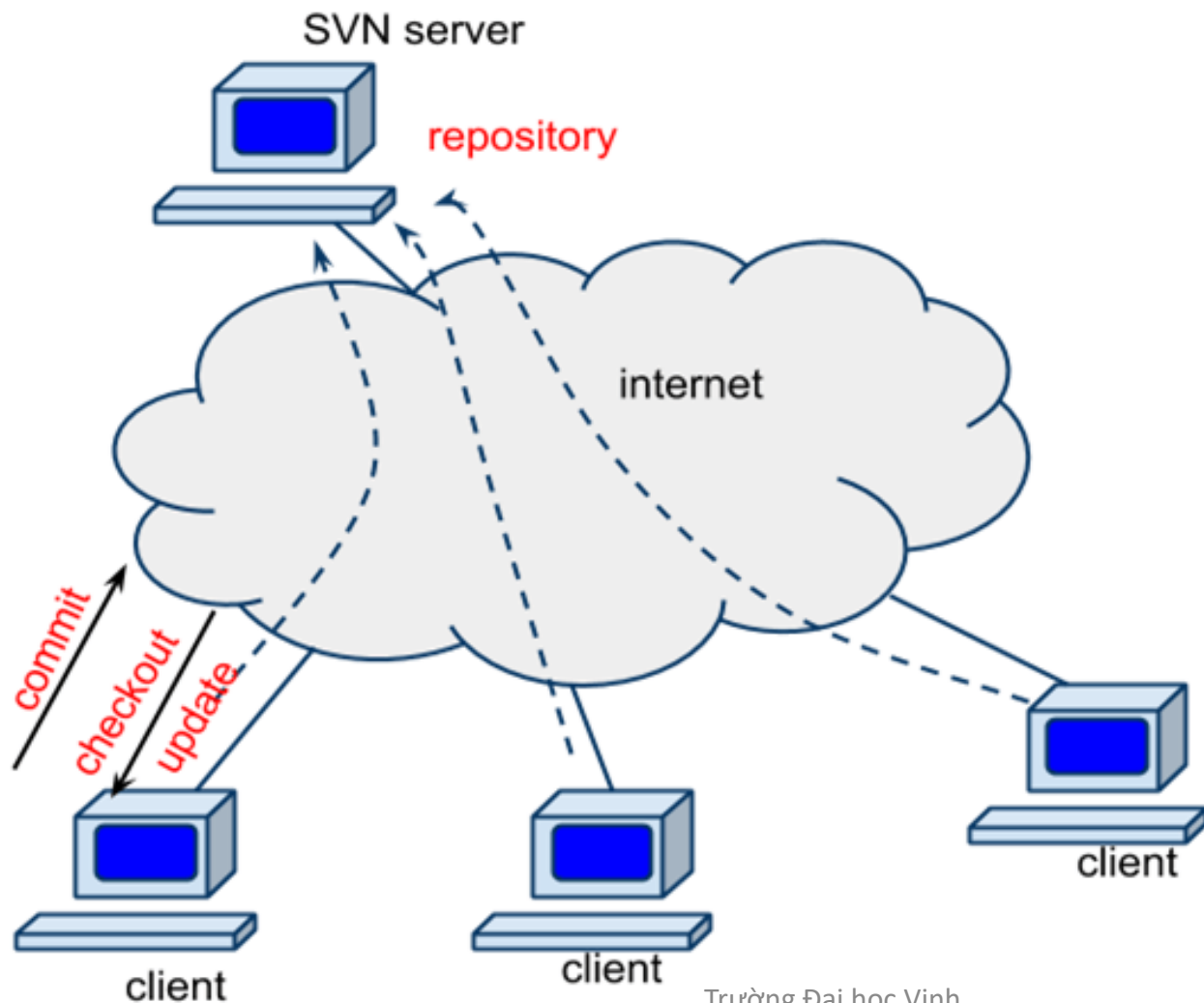
Hệ thống quản lý phiên bản mã nguồn

- Subversion (SVN): Một số đặc điểm
 - Là hệ thống quản lý source code tập trung (centralized)
 - Là hệ thống quản lý phiên bản mạnh mẽ, hữu dụng, và linh hoạt
 - Quản lý tập tin và thư mục theo thời gian
 - Giống như một hệ thống file server mà các client có thể download và upload file một cách bình thường
 - 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
 - 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



Hệ thống quản lý phiên bản mã nguồn

- Subversion (SVN):





Hệ thống quản lý phiên bản mã nguồn

- Subversion (SVN):
 - Visual SVN Server (dùng cho Server)
 - Tortoise SVN (dùng cho Client)
 - Sử dụng Eclipse và SVN
 - Sử dụng Visual Studio và SVN (AnkhSVN - plugin dành cho Visual Studio)



Hệ thống quản lý phiên bản mã nguồn

- Git:
 - Là phần mềm mã nguồn mở, có chức năng quản lý mã nguồn phân tán được phát triển bởi Linus Torvalds vào năm 2005
 - Là một trong các phần mềm quản lý mã nguồn phổ biến nhất
 - Phần mềm Git hỗ trợ việc cài đặt trên cả máy tính cá nhân và máy chủ chạy các hệ điều hành khác nhau như Mac OSX, Windows hay Linux
 - Có nhiều ưu điểm: tốc độ, đơn giản, phân tán, phù hợp với các dự án lớn nhỏ





Hệ thống quản lý phiên bản mã nguồn

- Git: Một số ưu điểm chính
 - Dễ sử dụng, an toàn và nhanh chóng
 - Quản lý source code dễ dàng chuyên nghiệp
 - Có thể giúp quy trình làm việc code theo nhóm đơn giản hơn rất nhiều bằng việc kết hợp các phân nhánh (branch)
 - Hạn chế được lỗi xảy ra trong quá trình code trong 1 team
 - Khi gặp lỗi có thể dễ dàng Backup lại phiên bản trước
 - Dễ dàng trong việc deployment sản phẩm





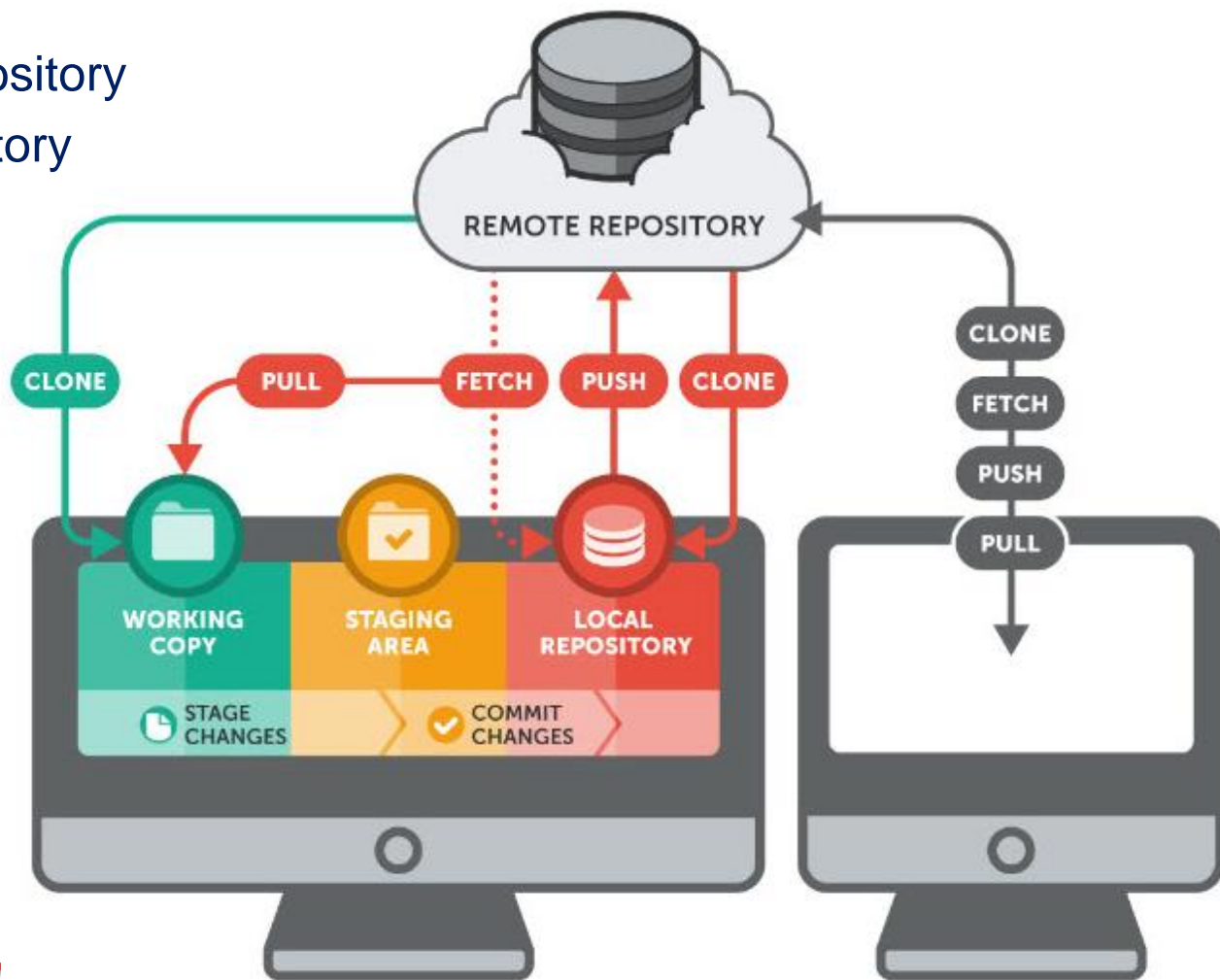
Hệ thống quản lý phiên bản mã nguồn

- Git: Một số đặc điểm
 - Các lưu trữ dữ liệu: Git lưu dữ liệu dưới dạng một loạt "ảnh chụp" (snapshot) của một tập hợp các file, mỗi khi commit (lưu lại) thì Git tiến hành chụp lại hệ thống các file thời điểm đó và lưu giữ một tham chiếu đến ảnh chụp đó
 - Hầu hết mọi thao tác với Git diễn ra ở Local: Vì mỗi máy trạm có một Database Git nên hầu hết thao tác với Git như thêm mới, xem lại lịch sử,... đều không cần đến Server (trừ cần commit lên Server, lấy về một file do người khác đã cập nhật)
 - Dữ liệu lưu trữ trong Git đảm bảo tính toàn vẹn: mọi thứ trước khi được lưu trữ vào Git đều được kiểm tra bởi mã băm (hash, checksum), nghĩa là không thể thay đổi nội dung của file mà Git không biết về sự thay đổi đó



Hệ thống quản lý phiên bản mã nguồn

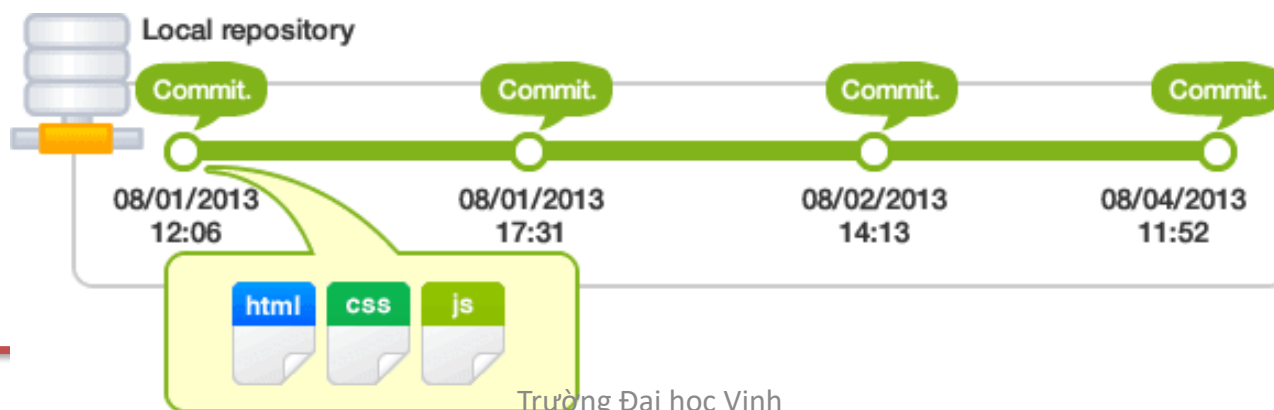
- Git: Một số khái niệm
 - Repository
 - Remote repository
 - Local repository
 - Branch
 - Merge
 - Conflict
 - Commit
 - Snapshot
 - Clone
 - Push
 - Fetch
 - Pull
 - ...





Hệ thống quản lý phiên bản mã nguồn

- Git: Các trạng thái
 - Committed: dữ liệu đã lưu trữ an toàn trong Database (local)
 - Modified: dữ liệu (file) có thay đổi nhưng chưa lưu trong Database (local)
 - Staged: đã đánh dấu các file sửa đổi modified chuẩn bị được commit tiếp theo





Hệ thống quản lý phiên bản mã nguồn

- Github:
 - Là một dịch vụ cung cấp kho lưu trữ mã nguồn Git dựa trên nền web cho các dự án phát triển phần mềm
 - Cũng là một mạng xã hội dành cho các lập trình viên, lập trình viên có thể theo dõi những người khác để xem hoạt động của họ, thêm bình luận cho các dự án phần mềm khác,...
 - Cung cấp các giao diện để người dùng có thể dễ dàng theo dõi các sự thay đổi trên trình duyệt mà không cần phải cài phần mềm Git trên máy tính
 - Nhiều nhà phát triển đã xem nó là một sự thay thế cho hồ sơ xin việc, một số nhà tuyển dụng đánh giá ứng viên thông qua kho mã nguồn của họ



GitHub



Hệ thống quản lý phiên bản mã nguồn

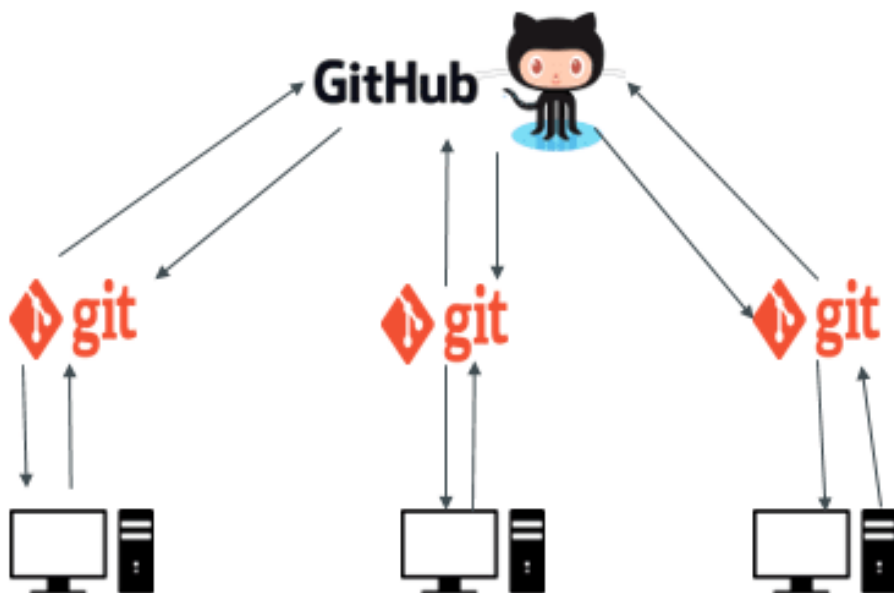
- Git/Github



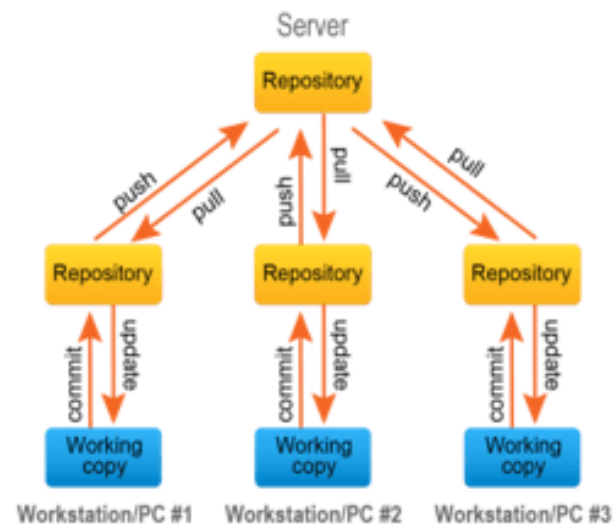
git



GitHub



Distributed version control





Câu hỏi thảo luận
