

Nhóm chuyên môn Nhập môn Công nghệ phần mềm

NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Quản lý phiên bản



(oxdot)

NỘI DUNG



- 1. Khái niệm quản lý phiên bản
- 2. Hệ thống kiểm soát phiên bản
- 3. Đánh số phiên bản

MŲC TIÊU



Sau bài học này, người học có thể:

- 1. Hiểu về các khái niệm của quản lý phiên bản phần mềm
- 2. Nắm được phân loại và đặc trưng của các hệ thống kiểm soát phiên bản
- 3. Hiểu về mô hình đánh số phiên bản phần mềm

NỘI DUNG TIẾP THEO



1. Khái niệm quản lý phiên bản

- 1.1. Quản lý phiên bản là gì?
- 1.2. Các khái niệm cơ bản
- 2. Hệ thống kiểm soát phiên bản
- 3. Đánh số phiên bản



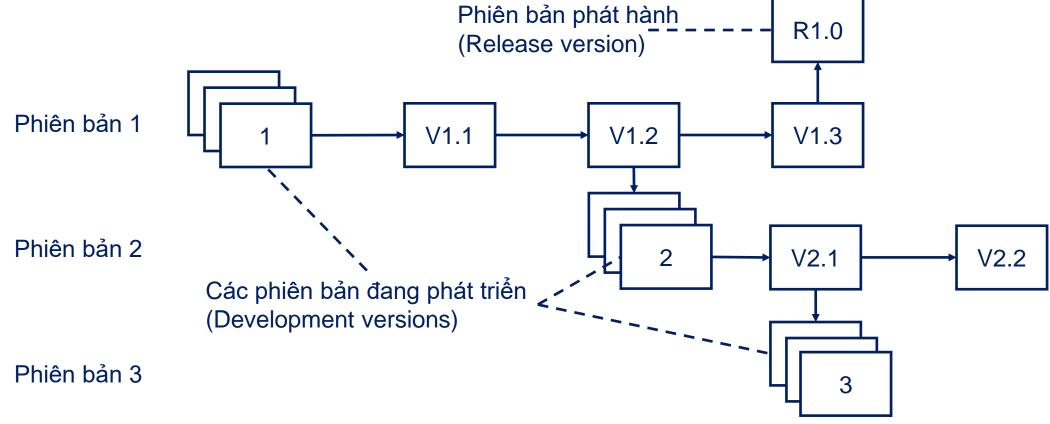
1.1. Quản lý phiên bản là gì?

- Quản lý phiên bản (Version management) hay kiểm soát phiên bản (Version control) là quá trình theo dõi các phiên bản khác nhau của các thành phần (các mục cấu hình) phần mềm và hệ thống mà các thành phần này được sử dụng.
- Đảm bảo các thay đổi do các nhà phát triển khác nhau thực hiện đối với các phiên bản không ảnh hưởng lẫn nhau.
- Quản lý phiên bản gắn liền với quản lý baseline, chỉ định các phiên bản thành phần được bao gồm trong hệ thống cụ thể.



1.1. Quản lý phiên bản là gì?

 Các phiên bản phần mềm khác nhau có thể bao gồm các thành phần chung cũng như các phiên bản thành phần khác biệt.



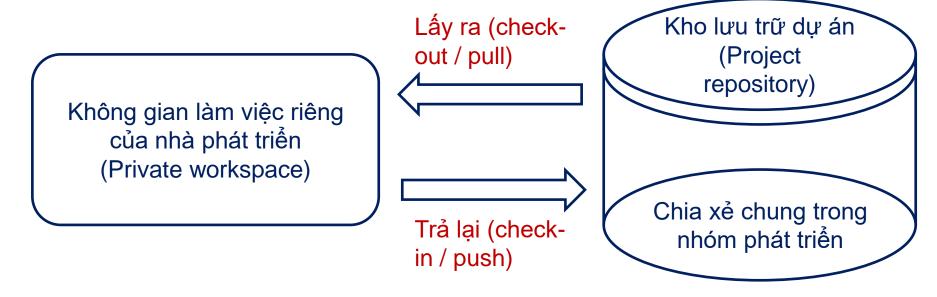
Hình 1.1: Phát triển phần mềm nhiều phiên bản



- Các hệ thống kiểm soát phiên bản sử dụng kho lưu trữ dự án (project repository) và không gian làm việc riêng (private workspace) để hỗ trợ phát triển độc lập, không can thiệp lẫn nhau.
- Kho lưu trữ dự án (project repository) duy trì phiên bản "chính" của tất cả các thành phần phần mềm. Nó được sử dụng để tạo đường cơ sở cho việc xây dựng hệ thống.
- Không gian làm việc riêng (private workspace) chứa bản sao của các thành phần từ kho lưu trữ. Nhà phát triển làm việc trên các bản sao này.



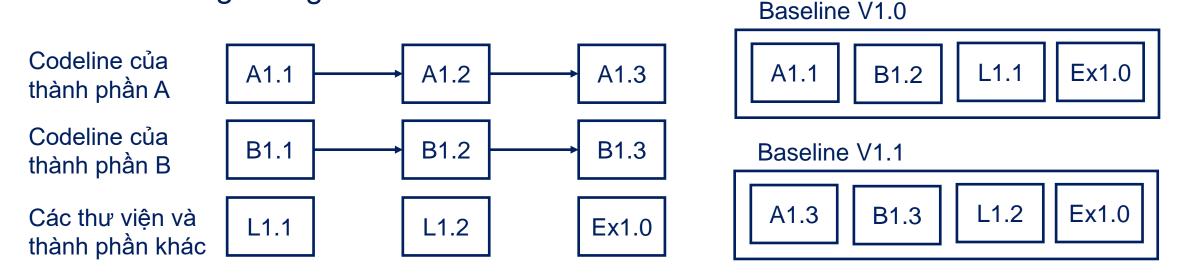
- Thao tác check-out / pull: nhà phát triển sao chép những thành phần cần sử dụng hoặc sửa đổi từ kho lưu trữ vào không gian làm việc riêng.
- Thao tác check-in / push: nhà phát triển hoàn thành các thay đổi của mình, các thành phần đã thay đổi sẽ được trả lại vào kho lưu trữ.



Hình 1.2: Thao tác với kho lưu trữ dự án và không gian làm việc riêng



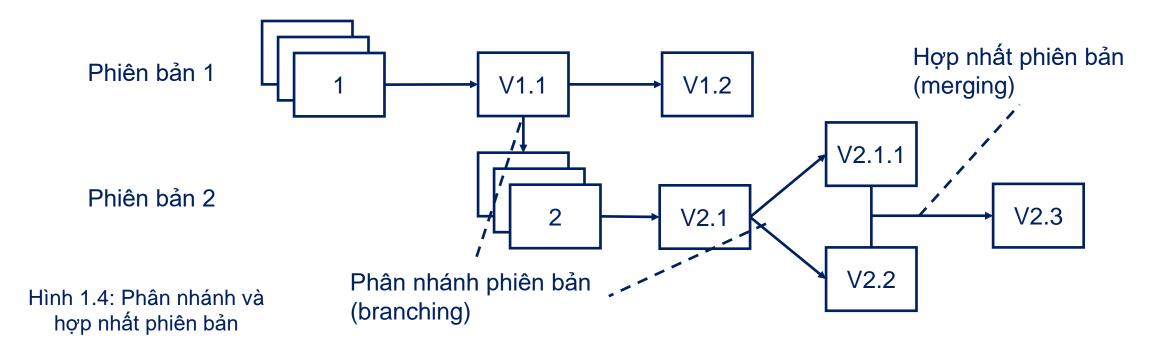
- Codeline là một chuỗi các phiên bản của một thành phần phần mềm. Trong đó, các phiên bản mới hơn trong chuỗi bắt nguồn từ các phiên bản trước đó.
- Baseline chỉ định các phiên bản thành phần (đã được phê duyệt) được bao gồm trong hệ thống và xác định các thư viện được sử dụng, các tệp cấu hình và những thông tin khác.



Hình 1.3: Codeline và Baseline



- Phân nhánh phiên bản (branching): sự phát triển độc lập của các lập trình viên trên cùng một thành phần tạo ra sự phân nhánh phiên bản.
- Hợp nhất phiên bản (merging): hợp nhất các nhánh phiên bản để tạo phiên bản mới của thành phần bao gồm tất cả các thay đổi đã được thực hiện.



NỘI DUNG TIẾP THEO



1. Khái niệm quản lý phiên bản

2. Hệ thống kiểm soát phiên bản

- 2.1. Đặc trưng của hệ thống kiểm soát phiên bản
- 2.2. Phân loại hệ thống kiểm soát phiên bản
- 2.3. Hệ thống kiểm soát phiên bản tập trung
- 2.4. Hệ thống kiểm soát phiên bản phân tán
- 3. Đánh số phiên bản



2.1. Đặc trưng của hệ thống kiểm soát phiên bản

- Định danh phiên bản và bản phát hành: gán mã định danh và tập thuộc tính để xác định duy nhất từng phiên bản.
- Ghi lại lịch sử thay đổi: lưu giữ hồ sơ về những thay đổi đã được thực hiện để tạo phiên bản mới của một thành phần từ phiên bản trước đó.
- Phát triển độc lập: kiểm soát các thay đổi đồng thời trên cùng một thành phần.
- Hỗ trợ phát triển dự án: hỗ trợ quản lý các thành phẩn được dùng chung giữa nhiều dự án
- Quản lý lưu trữ: sử dụng các cơ chế lưu trữ phiên bản hiệu quả.



2.2. Phân loại hệ thống kiểm soát phiên bản

- Hệ thống tập trung: có một kho lưu trữ chính duy nhất duy trì tất cả các phiên bản của các thành phần phần mềm đang được phát triển.
 - Subversion là một ví dụ về hệ thống kiểm soát phiên bản tập trung.
- Hệ thống phân tán: tồn tại nhiều phiên bản của kho thành phần cùng một lúc.
 - Git là một ví dụ về hệ thống kiểm soát phiên bản phân tán.



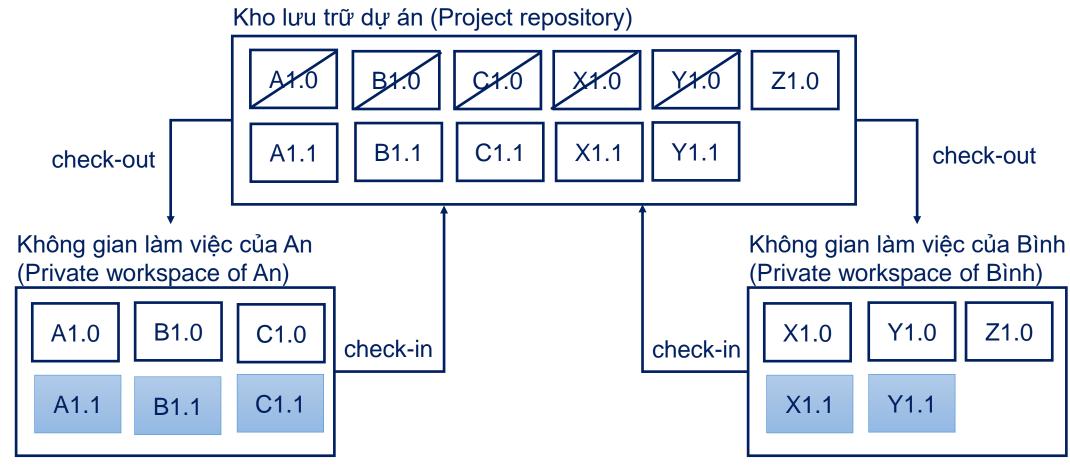
2.3. Hệ thống kiểm soát phiên bản tập trung

- Các nhà phát triển check-out các thành phần hoặc thư mục của các thành phần từ kho lưu trữ dự án vào không gian làm việc riêng và làm việc trên các bản sao của thành phần này.
- Khi các thay đổi được hoàn tất, nhà phát triển sẽ check-in các thành phần trở lại kho lưu trữ.
- Hệ thống kiểm soát phiên bản sẽ cảnh báo khi 2 người cùng check-out một thành phần.



2.3. Hệ thống kiểm soát phiên bản tập trung

Minh họa hệ thống kiểm soát phiên bản tập trung



Hình 2.1: Hệ thống kiểm soát phiên bản tập trung



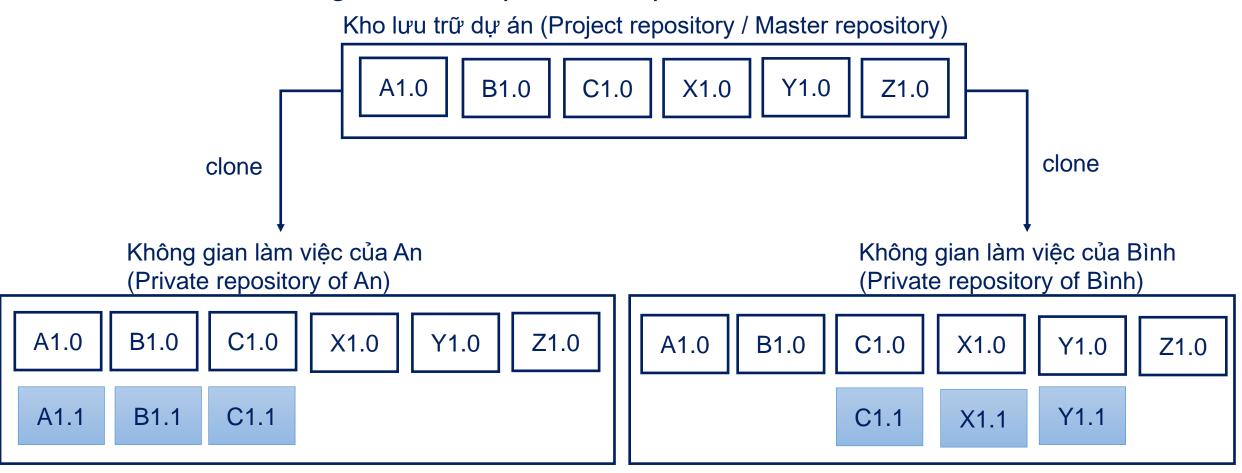
2.4. Hệ thống kiểm soát phiên bản phân tán

- Kho lưu trữ "master" được tạo trên máy chủ duy trì các sản phẩm do nhóm phát triển tạo ra.
- Thay vì chỉ check-out các thành phần cần dùng, một nhà phát triển tạo một
 bản sao của kho lưu trữ dự án (clone) vào không gian làm việc riêng.
- Các nhà phát triển làm việc trên các tệp được yêu cầu và duy trì các phiên bản mới trên kho lưu trữ riêng trên máy tính của mình.
- Khi các thay đổi được thực hiện, nhà phát triển "commit" những thay đổi này
 và cập nhật kho lưu trữ riêng.
- Sau đó, họ có thể "push" những thay đổi này vào kho dự án.



2.4. Hệ thống kiểm soát phiên bản phân tán

Minh họa hệ thống kiểm soát phiên bản phân tán



Hình 2.2: Tạo bản sao của kho lưu trữ trong Hệ thống kiểm soát phiên bản phân tán



2.4. Hệ thống kiểm soát phiên bản phân tán

- Uu điểm của hệ thống kiểm soát phiên bản phân tán:
 - Cung cấp một cơ chế sao lưu cho kho lưu trữ.
 - Nếu kho lưu trữ bị hỏng, công việc có thể tiếp tục và kho lưu trữ dự án có thể được khôi phục từ các bản sao cục bộ.
 - Cho phép hoạt động ngoại tuyến để các nhà phát triển có thể thực hiện các thay đổi nếu không có kết nối mạng.
 - Các nhà phát triển có thể biên dịch và thực thi toàn bộ hệ thống trên máy
 cục bộ để thử nghiệm những thay đổi đã thực hiện.

NỘI DUNG TIẾP THEO



- 1. Khái niệm quản lý phiên bản
- 2. Hệ thống kiểm soát phiên bản

3. Đánh số phiên bản

- 3.1. Khái niệm đánh số phiên bản
- 3.2. Các mô hình đánh số phiên bản



3.1. Khái niệm đánh số phiên bản

- Đánh số phiên bản hay định danh phiên bản phần mềm (Software versioning)
 - Một trong những hoạt động nền tảng của quản lý cấu hình.
 - Mục đích của định danh là để xác định tính duy nhất của một mục cấu hình và mối quan hệ của nó với các mục cấu hình khác.
 - Nó bao gồm việc mô tả tên, đánh số, đánh dấu đặc trưng, giúp nhận biết và phân biệt một mục cấu hình với các mục cấu hình hay thành phần khác.



3.2. Các mô hình đánh số phiên bản

- Một số phương pháp đánh số phiên bản được dùng phổ biến:
 - Đánh số phiên bản bằng các chuỗi số và ký tự (Sequence-based identifiers)
 - Đánh số phiên bản dựa theo các giai đoạn phát triển của sản phẩm (Stage-based identifiers)



3.2. Các mô hình đánh số phiên bản

- Đánh số phiên bản bằng các chuỗi số và ký tự
 - Công thức chung:

major.minor.[build [.revision]]
hoặc major.minor [maintenance[.build]]

- Ý nghĩa các số major, minor, build, revision
 - major: Chuỗi phiên bản chính
 - minor: Chuỗi phiên bản phụ
 - build: Chuỗi phiên bản cấu tạo. Đánh dấu sự khác nhau trong cùng
 1 phiên bản phụ, 2 chữ số
 - revision: Lần sửa đổi, đánh dấu lần sửa đổi của mã nguồn



3.2. Các mô hình đánh số phiên bản

- Đánh số phiên bản dựa theo các giai đoạn phát triển của sản phẩm
 - Các tên gọi của phiên bản:
 - Closebeta: Phiên bản thử nghiệm hạn chế
 - Openbeta: Phiên bản thử nghiệm diện rộng
 - Release Candidate (RC): phiên bản ứng viên
 - Official version: phiên bản chính thức

TỔNG KẾT VÀ GỢI MỞ



- 1. Bài học đã cung cấp cho người học một số **khái niệm cơ bản** về quản lý phiên bản phần mềm và **đặc trưng** của các hệ thống kiểm soát phiên bản
- 2. Tiếp sau bài này, **người học có thể tự tìm hiểu thêm** về các hệ thống quản lý phiên bản cụ thể được sử dụng trong thực tế



NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Quản lý phiên bản

Biên soạn:

ThS. Nguyễn Mạnh Tuấn

Trình bày:

ThS. Nguyễn Mạnh Tuấn





NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Bài học tiếp theo:

Ví dụ và bài tập: Quản lý phiên bản với hệ thống Git

Tài liệu tham khảo:

- [1] R. Pressman, Software Engineering: A Practitioner's Approach. 8th Ed., McGraw-Hill, 2016 và bộ slide đi kèm.
- [2] I. Sommerville, Software Engineering. 10th Ed., AddisonWesley, 2017.
- [3] Pankaj Jalote, An Integrated Approach to Software Engineering, 3rd Ed., Springer.
- [4] Shari Lawrence Pleeger, Joanne M.Atlee, Software Engineering theory and practice. 4th Ed., Pearson, 2009

Tư liệu:

- [1] git. Available at: https://git-scm.com/ (Accessed: 27 September 2023).
- [2] Subversion. Available at: https://subversion.apache.org/ (Accessed: 27 September 2023).

KẾ HOẠCH GIẢNG DẠY



Tuần	Chương	Video
5	Quy trình quản lý cấu hình phần mềm	C5.2-SCMProcess
5	Quản lý phiên bản	C5.3-VersionControl
5	Ví dụ và bài tập	C5.4-Ex