



ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Nhóm chuyên môn Nhập môn Công nghệ phần mềm

NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Khái niệm quản lý cấu hình phần mềm



1. Phần mềm và sự thay đổi
2. Quản lý cấu hình phần mềm
3. Các khái niệm

Sau bài học này, người học có thể:

1. Hiểu về **tổng quan** của sự thay đổi và sự cần thiết của kiểm soát thay đổi trong quy trình phát triển phần mềm
2. Hiểu về **nội dung** của quản lý cấu hình phần mềm
3. Nắm được **các khái niệm** trong quản lý cấu hình phần mềm

1. Phần mềm và sự thay đổi

1.1. Quá trình phát triển phần mềm

1.2. Sự tiến hoá của phần mềm

1.3. Những thay đổi của phần mềm

1.4. Thay đổi và kiểm soát

2. Quản lý cấu hình phần mềm

3. Các khái niệm

1. PHẦN MỀM VÀ SỰ THAY ĐỔI

1.1. Quá trình phát triển phần mềm

- Lý tưởng
 - Phần mềm được phát triển từ **các yêu cầu ổn định**
- Thực tế
 - Các yêu cầu ổn định luôn không tồn tại cho hầu hết các hệ thống thực tế
- Một dự án phần mềm hiệu quả cần phải có chiến lược để giải quyết vấn đề **“THAY ĐỔI”**

1. PHẦN MỀM VÀ SỰ THAY ĐỔI

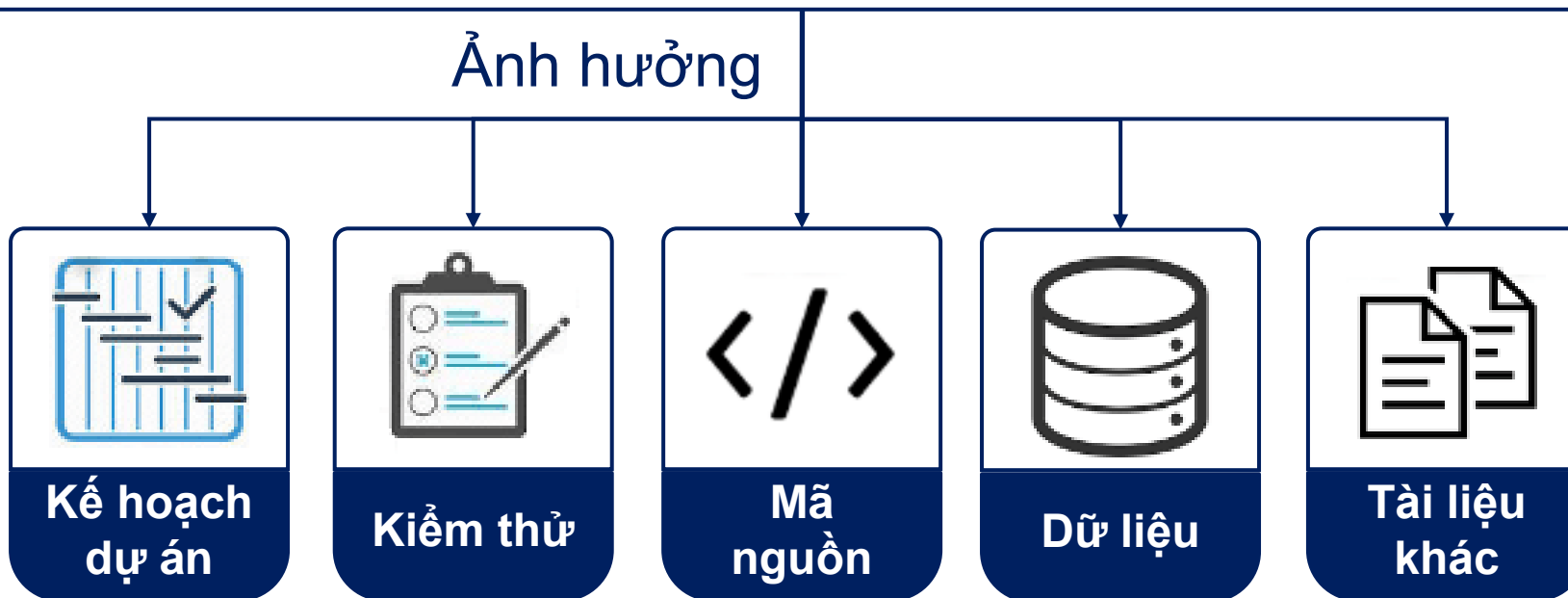
1.2. Sự tiến hóa của phần mềm

- Phần mềm được phát triển theo thời gian.
 - Nhiều yếu tố khác nhau được tạo ra trong suốt thời gian của dự án.
 - Có nhiều phiên bản khác nhau.
 - Các nhóm làm việc song song để đưa ra sản phẩm cuối cùng.
- Phần mềm có thể thay đổi liên tục.

1. PHẦN MỀM VÀ SỰ THAY ĐỔI

1.3. Những thay đổi của phần mềm

- Thay đổi yêu cầu nghiệp vụ
- Thay đổi yêu cầu kỹ thuật
- Thay đổi yêu cầu người dùng
- Thay đổi trong nhóm / tổ chức
- Thay đổi chính sách và luật lệ



1. PHẦN MỀM VÀ SỰ THAY ĐỔI

1.4. Thay đổi và kiểm soát

- Nếu những **thay đổi** không được **kiểm soát** - mọi thứ có thể và sẽ vượt khỏi tầm tay.
- Vấn đề quản lý thay đổi là cần thiết khi nhiều người cùng làm việc trong một dự án.
- Mọi **thay đổi** đều dẫn đến **rủi ro**.
 - Nếu không có các chiến lược và cơ chế thích hợp để kiểm soát các thay đổi → không thể khôi phục về bản sao cũ ổn định hơn của phần mềm.

1. PHẦN MỀM VÀ SỰ THAY ĐỔI



1.4. Thay đổi và kiểm soát

- Sự thật
 - Những thay đổi là không thể tránh khỏi
 - Các thay đổi cần được kiểm soát
 - Các thay đổi cần được quản lý
- Giải pháp
 - **Quản lý cấu hình phần mềm** - Software Configuration Management (SCM)

1. Phần mềm và sự thay đổi

2. Quản lý cấu hình phần mềm

2.1. Quản lý cấu hình phần mềm là gì?

2.2. Sự cần thiết của quản lý cấu hình phần mềm

3. Các khái niệm

2. QUẢN LÝ CẤU HÌNH PHẦN MỀM

2.1. Quản lý cấu hình phần mềm là gì?

- Quản lý cấu hình phần mềm - Software Configuration Management (SCM) bao gồm các nguyên tắc và kỹ thuật đánh giá và kiểm soát sự thay đổi đối với các sản phẩm phần mềm trong và sau quá trình kỹ thuật phần mềm.
- Các tiêu chuẩn (được ANSI phê duyệt)
 - IEEE 828: Software Configuration Management Plans
 - IEEE 1042: Guide to Software Configuration Management

2. QUẢN LÝ CẤU HÌNH PHẦN MỀM

2.1. Quản lý cấu hình phần mềm là gì?

- Áp dụng một cách tiếp cận nghiêm ngặt để đảm bảo:
 - **Các chi tiết** trong hệ thống phần mềm đều được xác định và theo dõi.
 - **Các thay đổi** với các mục khác nhau được ghi lại và theo dõi.
 - **Tích hợp** thích hợp tất cả các mô-đun khác nhau.

2. QUẢN LÝ CẤU HÌNH PHẦN MỀM

2.1. Quản lý cấu hình phần mềm là gì?

- SCM có thể giúp xác định tác động của thay đổi cũng như kiểm soát sự phát triển.
- Nó có thể theo dõi và kiểm soát các thay đổi trong tất cả các khía cạnh của phát triển phần mềm.
 - Yêu cầu
 - Mã hoá
 - Làm tài liệu
 - Thiết kế
 - Kiểm thử

2. QUẢN LÝ CẤU HÌNH PHẦN MỀM

2.2. Sự cần thiết của quản lý cấu hình phần mềm

- Khi phần mềm phát triển - nhiều tài nguyên hệ thống thay đổi.
 - SCM ngăn ngừa các lỗi có thể tránh được phát sinh từ các thay đổi xung đột.
- Thông thường nhiều phiên bản của phần mềm được phát hành và cần đến sự hỗ trợ.
 - SCM cho phép một nhóm hỗ trợ nhiều phiên bản.
 - SCM cho phép các thay đổi trong các phiên bản tuần tự được truyền bá.
- SCM cho phép các nhà phát triển theo dõi các thay đổi và khôi phục bất kỳ thay đổi nào để đưa hệ thống phần mềm trở lại trạng thái an toàn đã biết gần đây nhất.

1. Phần mềm và sự thay đổi
2. Quản lý cấu hình phần mềm

3. Các khái niệm

- 3.1. Mục cấu hình (Configuration Item - CI)
- 3.2. Đường cơ sở (Baseline)
- 3.3. Kho lưu trữ SCM (SCM Repository)
- 3.4. Thư mục SCM (SCM Directory)
- 3.5. Phiên bản, bản sửa đổi và bản phát hành (Version, Revision and Release)

3. CÁC KHÁI NIỆM



3.1. Mục cấu hình (Configuration Item - CI)

- Tập hợp phần cứng, phần mềm hoặc cả hai, được chỉ định để quản lý cấu hình và được coi như một thực thể duy nhất trong quy trình quản lý cấu hình.
- Các mục cấu hình phần mềm không chỉ là các đoạn mã chương trình mà là tất cả các loại tài liệu cho sự phát triển phần mềm.
- Ví dụ:
 - Các tệp mã nguồn
 - Tài liệu phân tích thiết kế
 - Trình điều khiển cho các trường hợp kiểm thử
 - Tài liệu hướng dẫn sử dụng
 - ...

3. CÁC KHÁI NIỆM



3.1. Mục cấu hình (Configuration Item - CI)

- Tìm kiếm các mục cấu hình
 - Các dự án lớn thường tạo ra hàng nghìn thực thể (tệp, tài liệu, ...) phải được **xác định duy nhất**.
 - Nhưng không phải tất cả các thực thể đều cần được định cấu hình.
- Các câu hỏi cần quan tâm
 - Nên quản lý những gì? - Lựa chọn mục cấu hình.
 - Khi nào bắt đầu đặt một thực thể dưới sự kiểm soát cấu hình?

3. CÁC KHÁI NIỆM

3.1. Mục cấu hình (Configuration Item - CI)

- Một số thực thể phải được duy trì trong suốt thời gian tồn tại của phần mềm.
 - Điều này cũng bao gồm giai đoạn khi phần mềm không còn được phát triển nhưng vẫn được sử dụng bởi khách hàng mong đợi sự hỗ trợ thích hợp trong nhiều năm.
- Một lược đồ đặt tên thực thể nên được xác định để định danh cho các tài liệu có tên liên quan.
- Lựa chọn các mục cấu hình phù hợp là một kỹ năng cần thực hành.
 - Sử dụng các kỹ thuật tương tự như mô hình hóa đối tượng để tìm các mục cấu hình.

3. CÁC KHÁI NIỆM

3.2. Đường cơ sở (Baseline)

- Đường cơ sở là đặc tả kỹ thuật hoặc sản phẩm đã được **xem xét** và **thống nhất chính thức**, sau đó được dùng như là một cơ sở để tiếp tục phát triển, và có thể thay đổi chỉ thông qua thủ tục kiểm soát thay đổi chính thức.
- Một baseline là một mốc quan trọng trong sự phát triển của phần mềm được đánh dấu bằng việc cung cấp một hoặc nhiều mục cấu hình phần mềm và sự chấp thuận của các mục cấu hình thu được thông qua đánh giá kỹ thuật chính thức.

3. CÁC KHÁI NIỆM



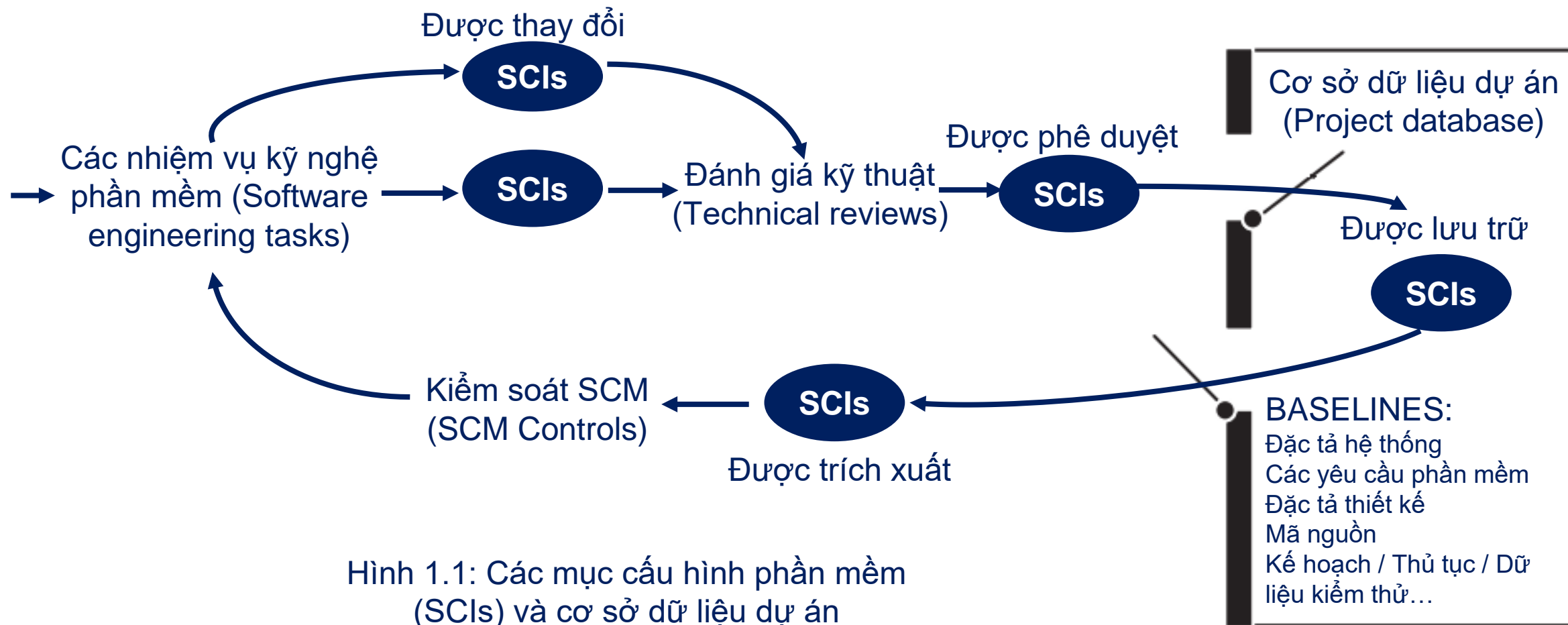
3.2. Đường cơ sở (Baseline)

- Baseline có thể được định nghĩa ở bất kỳ mức chi tiết nào
- Ví dụ:
 - Baseline A: API của một chương trình được xác định hoàn toàn, phần thân của các phương thức trống.
 - Baseline B: Tất cả các phương thức truy cập dữ liệu được thực hiện và kiểm thử, lập trình giao diện đồ họa người dùng có thể bắt đầu.
 - Baseline C: Giao diện đồ họa người dùng được triển khai, giai đoạn thử nghiệm có thể bắt đầu.

3. CÁC KHÁI NIỆM

3.2. Đường cơ sở (Baseline)

- Các mục cấu hình phần mềm (SCI) xác định đường cơ sở và cơ sở dữ liệu dự án



Hình 1.1: Các mục cấu hình phần mềm (SCIs) và cơ sở dữ liệu dự án

3. CÁC KHÁI NIỆM

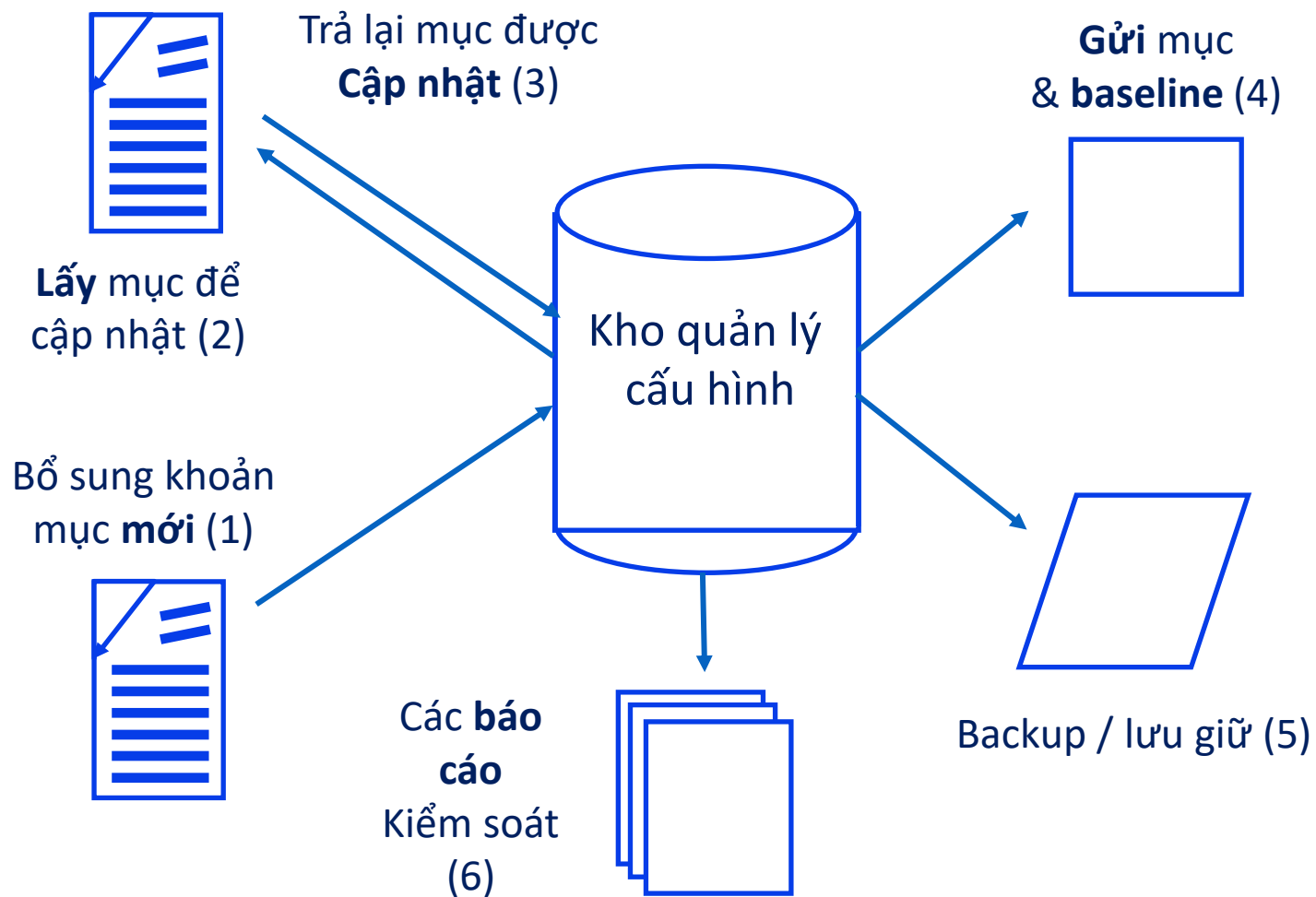


3.3. Kho lưu trữ SCM (SCM Repository)

- Kho lưu trữ SCM là một tập các cơ chế hoạt động và cấu trúc dữ liệu cho phép một nhóm phát triển phần mềm có thể quản lý thay đổi, phát triển, bảo trì phần mềm một cách hiệu quả.
- Một kho lưu trữ (repository) có các chức năng:
 - Toàn vẹn dữ liệu
 - Tích hợp dữ liệu
 - Chia sẻ thông tin
 - Thực thi phương pháp luận
 - Tích hợp công cụ
 - Tiêu chuẩn hoá tài liệu

3. CÁC KHÁI NIỆM

3.3. Kho lưu trữ SCM (SCM Repository)



Hình 1.2: Kho lưu trữ SCM và các hoạt động

3. CÁC KHÁI NIỆM



3.4. Thư mục SCM (SCM Directory)

- Programmer's Directory (IEEE: Dynamic Library)
 - Thư viện để chứa các thực thể phần mềm mới được tạo hoặc sửa đổi. Không gian làm việc của lập trình viên chỉ do lập trình viên kiểm soát.
- Master Directory (IEEE: Controlled Library)
 - Quản lý (các) baseline và kiểm soát các thay đổi được thực hiện đối với chúng. Mục nhập được kiểm soát, thường sau khi được xác minh. Các thay đổi phải được cho phép.
- Software Repository (IEEE: Static Library)
 - Lưu trữ cho các baseline khác nhau được phát hành để sử dụng chung. Các bản sao của các baseline này có thể được cung cấp cho các tổ chức yêu cầu.

3. CÁC KHÁI NIỆM



3.5. Phiên bản, bản sửa đổi và bản phát hành (Version, Revision & Release)

- Phiên bản (Version)
 - Một bản phát hành **ban đầu** hoặc tái phát hành một mục cấu hình được liên kết với một bản biên dịch hoặc biên dịch lại **hoàn chỉnh** của mục đó. Các phiên bản khác nhau có chức năng khác nhau.
- Bản sửa đổi (Revision)
 - **Thay đổi** thành phiên bản chỉ sửa các lỗi trong thiết kế / mã, nhưng không ảnh hưởng đến chức năng đã được lập thành tài liệu.
- Bản phát hành (Release)
 - Việc **phân phối chính thức** phiên bản đã được phê duyệt.

1. Bài học đã cung cấp cho người học một số **khái niệm cơ bản** trong quản lý cấu hình phần mềm.
2. Tiếp sau bài này, **người học có thể tự tìm hiểu thêm** về các hoạt động trong quy trình quản lý cấu hình phần mềm.

NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Khái niệm quản lý cấu hình phần mềm

Biên soạn:

ThS. Nguyễn Mạnh Tuấn

Trình bày:

ThS. Nguyễn Mạnh Tuấn



NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Bài học tiếp theo:

Quy trình quản lý cấu hình phần mềm

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 Pleege, Joanne M. Atlee, Software Engineering theory and practice. 4th Ed., Pearson, 2009

Tư liệu:

- [1] 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology. Available at: <https://ieeexplore.ieee.org/document/159342> (Accessed: 27 September 2023).