

BÀI TẬP THỰC HÀNH SỐ 2

THỰC HÀNH QUẢN LÝ CẤU HÌNH

1. MỤC TIÊU

Sau khi kết thúc bài thực hành, các nhóm phải hoàn thành các nhiệm vụ sau:

- Lập kế hoạch quản lý cấu hình (xác định các mục cấu hình, quy trình kiểm soát thay đổi, ...)
- Cài đặt môi trường làm việc, phân quyền truy cập và lưu trữ các file theo đúng cấu trúc thư mục đã định trong Kế hoạch quản lý cấu hình
- Mô hình hóa yêu cầu nghiệp vụ (viết báo cáo theo mẫu Lab 2 của CNPM, bao gồm cả Thiết kế kiến trúc hệ thống)

2. CƠ SỞ LÝ THUYẾT

2.1. Lập kế hoạch quản lý cấu hình (QLCH)

Thông thường, việc lập kế hoạch QLCH được thể hiện trong tài liệu có tên Kế hoạch quản lý cấu hình (Configuration Management Plan – CMP).

Bản kế hoạch này thường bao gồm các phần sau:

1. Ý nghĩa, mục đích và phạm vi áp dụng của bản kế hoạch
2. Vai trò và trách nhiệm
 - Cá nhân trong dự án thực hiện các hoạt động khác nhau liên quan đến QLCH.
 - Định nghĩa rõ ràng ai thực hiện (perform), ai xem xét (review), ai phê duyệt (approve) trên các CI của dự án, cũng như vai trò của khách hàng, người sử dụng đầu cuối.
3. Công cụ (tool), môi trường (environment) và cơ sở hạ tầng (infrastructure).
 - Phần này mô tả các công cụ phần mềm hoặc quy trình thủ tục được sử dụng hỗ trợ QLCH, chẳng hạn công cụ quản lý phiên bản sản phẩm (version control); mô tả vị trí các máy chủ, máy trạm, cấu hình hệ thống client-server, ...
4. Phương pháp định danh và thiết lập baseline trên các CI
5. Quy ước đặt tên trong dự án, gồm cả tên file
6. Quy trình xử lý và quản lý các thay đổi (change control process)
7. Chỉ định thành viên nhóm Configuration Control Board (CCB)
8. Thông tin nơi lưu trữ các CI
9. Kiểm kê và báo cáo cấu hình (configuration accounting and reporting)
10. Quy trình thủ tục lưu trữ và sao chép dự phòng (backup and archive)

2.2. Biểu mẫu

Sử dụng biểu mẫu NASA_SwCMPlanTemplate.doc

Và điền hoàn thiện các phần 2.0, 3.0 và 4.1, 4.2, 4.3

2.3. Vai trò và trách nhiệm (Project Roles and Responsibilities)

Trong một dự án điển hình, thông thường có 4 (nhóm) chức năng sau (thường gọi tắt là role) tham gia thực hiện các hoạt động QLCH:

CM (Configuration manager)

- Thiết lập và bảo trì kho lưu trữ (repository) của dự án.
- Phát triển và triển khai các quy trình thủ tục QLCH của dự án.
- Thiết lập các baseline, ghi nhận chi tiết các thay đổi trên các baseline
- Bảo đảm các baseline không bị thay đổi khi chưa được phê chuẩn.
- Tổ chức và điều phối các cuộc họp của CCB

Trưởng dự án (Project manager):

- Giám sát các hoạt động QLCH.
- Bảo đảm các yêu cầu cần thiết cho hoạt động QLCH. Ví dụ: số giờ thành viên dự án bỏ ra cho QLCH, công cụ hỗ trợ cho QLCH...

CCB (Configuration Control Board)

Bao gồm các thành viên trong dự án, và có chức năng như đã nói ở trên.

Các thành viên của dự án

Các thành viên của dự án, kể cả CM, PM và thành viên CCB, có trách nhiệm:

- Tuân thủ tất cả các quy trình thủ tục của bản kế hoạch QLCH (CMP)
- Tham gia vào nhóm CCB khi có yêu cầu

2.4. Resources and Environment**2.4.1. Công cụ quản lý cấu hình**

Các nhóm có thể sử dụng công cụ dưới đây để quản lý cấu hình

- Server: <https://bitbucket.org/>
- Client: Git/Mercurial + Source Tree

Tham khảo file hướng dẫn cách sử dụng Bitbucket + Mercurial (Bitbucket_WithMercurial_guide)

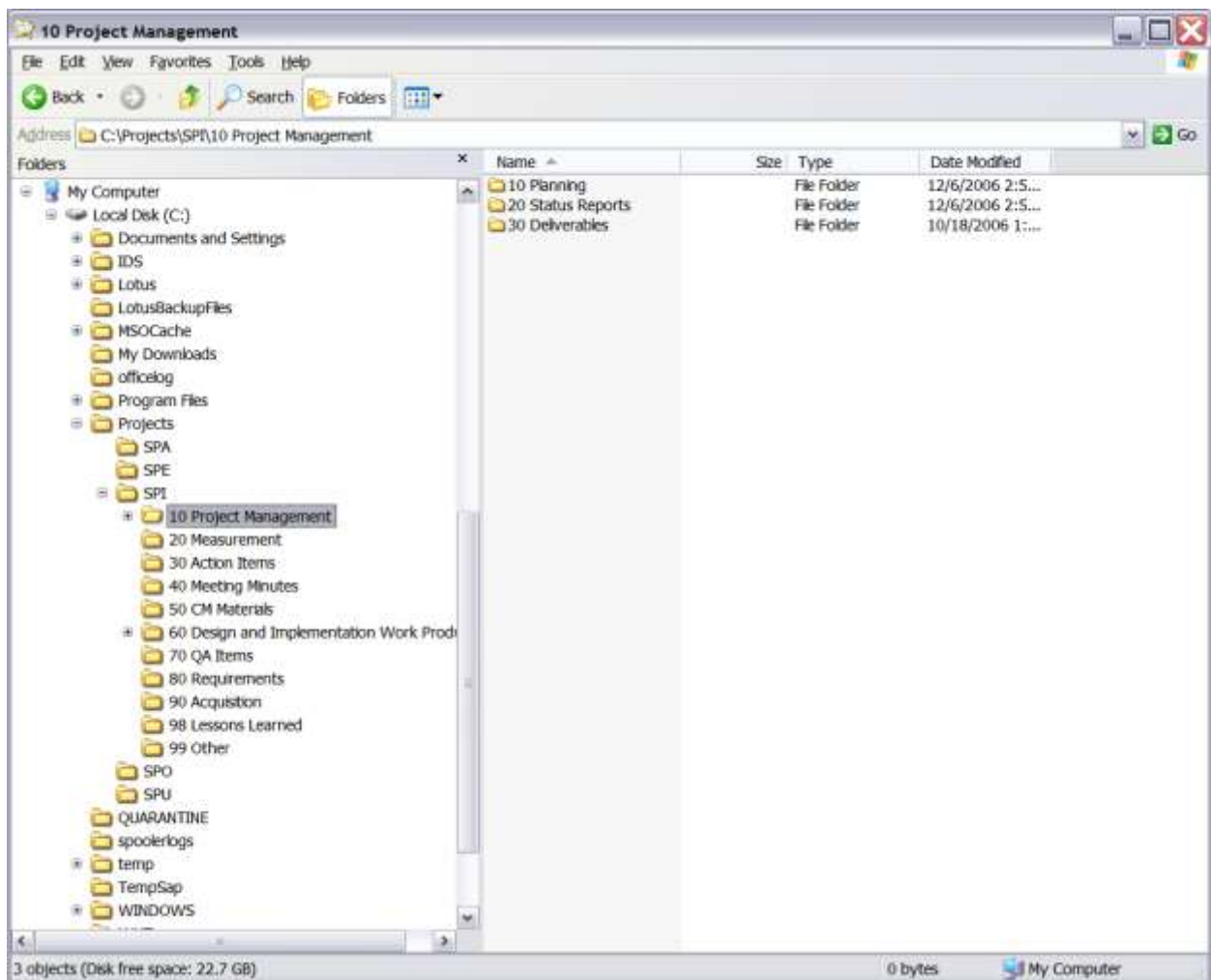
2.4.2. Cấu trúc thư mục lưu trữ (Repository)

Project Name = Gxy_Tên-phần-mềm

Trong Bitbucket, tạo các thư mục với cấu trúc như hình dưới đây:

- 10 Project Management
 - Project Plan: kế hoạch dự án hoặc kết nối sang tool quản lý dự án ASANA
 - Project Report: báo cáo tiến độ dự án (tuần, tháng, version, ...)
 - Project Estimation: Function Point, Cocomo
- 20 Measurement
 - Actual vs Plan
 - Bug Report: file excel báo cáo lỗi theo từng version / release (tổng hợp từ Issue trong Bitbucket)
 - Productivity
- 30 Action Items
 - OT Plan
 - Reschedule Plan
- 40 Meeting Minutes
- 50 CM Materials
 - CM Plan

- Setup
- Guidelines (Coding Convention, Design Guideline, ...)
- 60 Realisation
 - A&D
 - Test
 - Source Code
- 70 QA Items: báo cáo Audit
- 80 Requirements
 - SRS (Software Requirement Specification)
 - Templates: biểu mẫu, quy định
- 90 Acquisition
 - 3rd part: Software, Component
 - Outsourcing Plan
- 98 Lesson Learned: các bài học kinh nghiệm nhóm rút ra sau mỗi lần release
- 99 Other



2.5. Định danh/đánh số các CI (Identification of Configuration Items)

Định danh là một trong những hoạt động nền tảng của QLCH. Mục đích của định danh là để xác định tính duy nhất của một CI, cũng như mối quan hệ của nó với các CI khác. Nó bao gồm việc mô tả tên, đánh số, đánh dấu đặc trưng, giúp nhận biết và phân biệt một CI với các CI hay thành phần khác.

Trong sản xuất phần mềm, một CI có thể bao gồm một hay nhiều file. Ví dụ: một module tên ExpMod có thể được coi là một CI, module này có 2 file ExpMod.h và ExpMod.c.

Mỗi CI phải có một số định danh duy nhất, dạng thức thường thấy là:

Ví dụ: *PRJ001_REQB_1.0.4_draft_B* cho biết:
 Số ID của dự án: *PRJ001*
 Số ID của Item : *REQB*
 Phiên bản: *1.0.4_draft_B*

Trong một dự án, thường có rất nhiều file source code, quy tắc cơ bản là: các file cùng tạo nên một khối chức năng được gom chung thành một CI.

2.6. Kiểm soát phiên bản (Version Control)

Version control là sự kiểm soát các phiên bản (version) khác nhau của một CI (bao gồm việc định danh và lưu trữ CI đó).

Phiên bản là một thực thể mới của một CI sau khi đã qua một hoặc nhiều lần xem xét và thay đổi. Mỗi phiên bản sẽ có một số ID đầy đủ, và được tăng dần cho mỗi phiên bản mới. Lưu ý rằng phiên bản của một CI khác với phiên bản của các file thành phần của CI đó.

2.6.1. Hệ thống quản lý phiên bản (Revision/Version Control System)

Hiện có nhiều công cụ trên thị trường hỗ trợ cho việc kiểm soát phiên bản, một số công cụ thông dụng là: Visual Source Safe của Microsoft, ClearCase của Rational, CVS, Subversion, Github (nguồn mở).

Đọc thêm

- Tài liệu về [git](#)
- Tài liệu về [mercurial](#)

Các Hệ thống Quản lý phiên bản (RCS) cung cấp **đồng thời** ba chức năng quan trọng nhất:

- **Chức năng phục hồi:** cho phép khôi phục trạng thái trước khi phát hiện ra có lỗi trong một thao tác sửa đổi đã làm.
- **Chức năng đồng bộ:** cho phép nhiều người cùng sửa một/một tập hợp file cùng lúc vì các thao tác sửa đổi gây xung đột có thể được phát hiện và giải quyết sau đó.
- **Ghi lại lịch sử:** ghi lại các thông tin cần thiết của từng sửa đổi một, như tác giả, thời gian, các ghi chú giải trình thao tác thay đổi đó v.v...

2.6.2. Các thao tác cơ bản của RCS

Commit

- Commit là thao tác ra lệnh cho RCS ghi lại những thay đổi (change-sets) mà tác giả vừa thực hiện với (các) file mà RCS đang theo dõi. Commit cũng được dùng với nghĩa là những thay đổi nằm trong cùng một lần “commit” (ra lệnh) này. “Những thay đổi” ở đây có thể là tạo file, đổi tên/di chuyển, xoá file, sửa đổi nội dung v.v...
- Để các commit có giá trị thông tin (xem lại sau này chẳng hạn), ngoài thông tin tác giả và thời gian được RCS tự động thêm vào, người sử dụng phải nhập các ghi chú (comment) lí giải về commit đó. Các ghi chú của commit được gọi là “commit message”. Các message nên ngắn gọn, giải thích xem những sửa đổi đó nhằm làm gì (và khái quát cách làm nếu nó phức tạp).
- Khi cần xem lại các commit trước đó, người ta gọi một cái gọi là “commit history” hay lịch sử commit.

Thao tác sửa đổi nội dung: patch và diff

- So với việc lưu lại từng phiên bản, việc **chỉ lưu những phần thay đổi** có lợi hơn cho không gian đĩa, cũng như giúp cho việc trao đổi các thay đổi trong môi trường mạng máy tính nhanh hơn. Thao tác “diff” sẽ tự động **phân tích hai phiên bản khác nhau** của một file để chỉ ra phần khác nhau (diff). Phần khác nhau này có thể được xuất ra thành một file gọi là “patch” chứa **thông tin về thay đổi theo quy chuẩn**.
- Cần lưu ý là patch với nghĩa là “bản vá” chỉ là một ứng dụng cụ thể của patch. Thao tác applying patch nói chung là “từ file trước sửa đổi và bản patch tổng hợp thành file sau sửa đổi”.
- Các thao tác diff và patch có thể thực hiện trực tiếp thông qua các lệnh cùng tên trên các hệ thống GNU/Linux hoặc gián tiếp nhờ sự giúp đỡ của các RCS.

Xung đột (conflict) giữa các commit

- Xung đột (conflict) giữa các commit xảy ra khi có hai (hay nhiều hơn) commit độc lập với nhau xuất phát từ cùng một commit cũ trước đó. Xung đột thường xảy ra khi có một trong các commit mới đó chứa thao tác xóa bỏ. Trường hợp các commit đều là thao tác thêm sẽ dễ dàng được các RCS tự động xử lý (resolve). Khi xung đột xảy ra, lập trình viên phải cho RCS biết kết quả cuối cùng mà mình mong muốn.

2.7. Kiểm soát thay đổi (Change control)

Khi phát triển hoặc bảo trì một sản phẩm phần mềm, việc thay đổi yêu cầu là không thể tránh khỏi. Mục đích của change control là để kiểm soát đầy đủ tất cả các thay đổi ảnh hưởng đến việc phát triển một sản phẩm. Đôi lúc chỉ một vài yêu cầu thay đổi nhỏ của khách hàng, tất cả các chặng của quy trình phát triển phần mềm từ thiết kế, đến viết code, đến kiểm tra sản phẩm đều phải thay đổi theo.

Yêu cầu trong kiểm soát thay đổi là mọi sự thay đổi phải được thông báo đến tất cả những người hoặc nhóm làm việc có liên quan.

Các bước cơ bản của kiểm soát thay đổi bao gồm:

- Nghiên cứu, phân tích
- Phê chuẩn hoặc không phê chuẩn
- Thực hiện việc thay đổi
- Kiểm tra việc thay đổi
- Xác lập baseline mới

Trong kiểm soát thay đổi, ta cũng thường gặp khái niệm “nhóm kiểm soát thay đổi” gọi tắt là CCB (Change Control Board), nhóm này được thành lập trong từng dự án. CCB thông thường bao gồm:

- Người QLC H (Configuration Manager)
- Trưởng dự án (Project Manager)
- Trưởng nhóm (Technical Lead)
- Trưởng nhóm kiểm soát chất lượng (Test Lead)
- Kỹ sư chất lượng (Quality Engineer)
- Và những ai bị ảnh hưởng bởi các thay đổi

Nhiệm vụ của CCB thường là:

- Bảo đảm tất cả các thay đổi là được các bộ phận liên quan nhận biết và tham gia
- Xem xét, phê chuẩn hoặc phủ quyết các thay đổi trên các baseline
- Kiểm tra, xác nhận các thay đổi
- Phê chuẩn các bản phân phối sản phẩm (release) đến khách hàng

2.8. Báo cáo tình trạng cấu hình (Configuration Status Accounting)

Công việc này bao gồm việc ghi nhận và báo cáo tình trạng của các CI cũng như yêu cầu thay đổi, tập hợp số liệu thống kê về CI, đặc biệt là các CI góp phần tạo nên sản phẩm. Nó trả lời những câu hỏi như: có bao nhiêu file bị ảnh hưởng khi sửa chữa một lỗi phần mềm nào đó?

Kết quả của công việc này được ghi nhận trong một báo cáo mang tên Configuration Status Accounting Report (CSAR). Báo cáo này thường làm rõ những điểm sau:

- Liệt kê tất cả baseline và CI thành phần hoặc có liên quan.
- Làm nổi bật các CI đang được phát triển hoặc vừa bị thay đổi
- Liệt kê các thay đổi còn đang dang dở hay đang hoàn thành, và các baseline bị ảnh hưởng (bởi sự thay đổi đó)

Việc báo cáo này được làm thường xuyên và định kỳ, xuyên suốt dự án.

2.9. Auditing

Có 3 loại audit thường được thực hiện.

- CSAR Audit: Thường được làm sau mỗi lần một CSAR được tạo ra, việc kiểm tra bao gồm:
 - Bảo đảm các baseline mới nhất được liệt kê trong CSAR
 - Bảo đảm tất cả CI tạo nên một baseline được liệt kê
 - Kiểm tra các CI đã bị thay đổi từ lần baseline trước đó, so sánh chúng với các yêu cầu thay đổi để khẳng định rằng sự thay đổi trên CI là hợp lý.
- Physical configuration audit (PCA): nhằm mục đích khẳng định xem những gì khách hàng yêu cầu có được hiện thực hay không. Gồm 2 việc:
 - Kiểm tra vết để phản ánh tính 2 chiều (traceability) giữa yêu cầu khách hàng và việc hiện thực code trong dự án.
 - Xác định những gì sẽ được phân phối cho khách hàng (executable files, source code, tài liệu đi kèm...) có đáp ứng yêu cầu khách hàng hay không.
- Functional configuration audit (FCA): nhằm mục đích khẳng định những gì khách hàng yêu cầu có được kiểm tra chặt chẽ trên sản phẩm tạo ra trước khi giao cho khách hàng hay không. Gồm:
 - Kiểm tra vết để phản ánh tính 2 chiều giữa yêu cầu khách hàng và việc kiểm tra sản phẩm.

2.10. Quản lý release (Release management)

Trong thực tế, có nhiều định nghĩa khác nhau về khái niệm “release”. Về cơ bản, chúng ta có thể hiểu: Quá trình phát triển một phần mềm thường qua nhiều lần tích hợp, kết quả của mỗi lần tích hợp là một bản “build”, trong rất nhiều bản “build” đó, một số bản đáp ứng một số yêu cầu đã định hoặc lập kế hoạch trước (theo yêu cầu khách hàng), sẽ được gửi cho khách hàng để kiểm tra hoặc đánh giá. Các bản build này được gọi là “release”; công việc tạo ra và phân phối các bản release được gọi là công việc “release”. Theo cách hiểu này, sản phẩm sau cùng cũng là một bản release, đôi khi được gọi là “final release”.

Trong quá trình release, việc quản lý đòi hỏi phải thực hiện các công việc sau:

- Baseline môi trường phát triển sản phẩm và các file, tài liệu (sẽ release)
- Thực hiện báo cáo CSAR (xem định nghĩa ở trên)
- Thực hiện các audit: PCA và FCA
- Đóng gói file và tài liệu sẽ release
- Xác nhận đã nhận bản release từ khách hàng

2.11. Lưu trữ và chép dự phòng (Backup & archive)

Lưu trữ và chép dự phòng là một hoạt động của QLCH và là một trong những hoạt động quan trọng phải có của sản xuất phần mềm. Nó giúp khắc phục các trường hợp rủi ro bị mất mát dữ liệu do thao tác sai, virus, hoặc sự cố phần cứng/ phần mềm. Ở khía cạnh khác, nó hỗ trợ cho hoạt động version control trong trường hợp ta muốn sử dụng những version khác nhau.

Lưu trữ và chép dự phòng đòi hỏi toàn bộ sản phẩm và sản phẩm trung gian của dự án phải được định kỳ chép dự phòng trên những thiết bị hoặc những nơi khác một cách an toàn.

Và khi dự án kết thúc, các hoạt động sau cần phải thực hiện:

- Lưu trữ toàn bộ dữ liệu của dự án, tuân thủ quy trình lưu trữ đã được thiết lập (định nghĩa bởi dự án hoặc quy định ở cấp công ty).
- Lưu trữ hoặc huỷ bỏ các tài liệu ở dạng giấy.
- Dọn sạch dữ liệu hoặc thông tin của dự án vừa kết thúc, sau khi đã chép lưu trữ.