CHƯƠNG 6 QUẢN LÝ CẤU HÌNH - Configuration management

1 Tổng quan – OVERVIEW

* 1. QUẢN LÝ CẤU HÌNH - Configuration management
* Hệ thống phần mềm liên tục thay đổi trong quá trình phát triển vàsử dụng.
* Quản lý cấu hình (CM) quan tâm đến các chính sách, các quy trình và công cụ quản lý hệ thống phần mềm đang thay đổi.
* Bạn cần CM vì rất dễ mất dấu những thay đổi và các phiên bản thành phần đã được tích hợp vào mỗi phiên bản hệ thống.
* CM là cần thiết cho các dự án nhóm để kiểm soát các thay đổi do các các nhà phát triển.
  1. Thuật ngữ CM

|  |  |
| --- | --- |
| Mục cấu hình hoặc mục cấu hình phần mềm (SCI) | Bất kỳ thứ gì liên quan đến một dự án phần mềm (thiết kế, mã, dữ liệu thử nghiệm, tài liệu, v.v.) đã được đặt dưới sự kiểm soát cấu hình. Thường có các phiên bản khác nhau của một mục cấu hình. Các mục cấu hình có  tên riêng. |
| Kiểm soát cấu hình | Quá trình đảm bảo rằng các phiên bản của hệ thống và thành phần được ghi lại và duy trì để các thay đổi được quản lý và tất cả các phiên bản của thành phần được xác định và lưu trữ trong suốt thời gian tồn tại của hệ thống. |
| Phiên bản | Một ví dụ của mục cấu hình, theo một cách nào đó, khác với các phiên bản khác của mục đó. Các phiên bản luôn có một số nhận dạng duy nhất, thường bao gồm tên mục cấu hình cộng với số phiên bản. |
| Đường cơ sở | Đường cơ sở là một tập hợp các phiên bản thành phần tạo nên một hệ thống. Đường cơ sở được kiểm soát, có nghĩa là không thể thay đổi phiên bản của các thành phần tạo nên hệ thống. Điều này có nghĩa là luôn có thể tạo lại đường cơ sở từ các thành phần cấu thành của nó |
| Codeline | Dòng mã là một tập hợp các phiên bản của một thành phần phần mềm và các mục cấu hình khác mà thành phần đó phụ thuộc vào. |
| Dòng chính | Một chuỗi các đường cơ sở đại diện cho các phiên bản khác nhau của hệ thống. |
| Phóng thích | Một phiên bản của hệ thống đã được phát hành cho khách hàng (hoặc những người dùng khác trong một tổ chức) để sử dụng. |
| Không gian làm việc | Một khu vực làm việc riêng nơi phần mềm có thể được sửa đổi mà không ảnh hưởng đến các nhà phát triển khác, những người có thể đang sử dụng hoặc sửa đổi phần mềm đó. |
| Sự phân nhánh | Việc tạo dòng mã mới từ một phiên bản trong dòng mã hiện có. Sau đó, dòng mã mới và dòng mã hiện có có thể phát triển độc lập. |
| Hợp nhất | Việc tạo ra một phiên bản mới của một thành phần phần mềm bằng cách hợp nhất các phiên bản riêng biệt theo các nguyên tắc khác nhau. Những nguyên tắc này có thể có  được tạo bởi một nhánh trước của một trong các quy tắc liên quan. |
| Xây dựng hệ thống | Việc tạo ra một phiên bản hệ thống thực thi bằng cách biên dịch và liên kết các phiên bản thích hợp của các thành phần và thư viện tạo nên hệ thống. |

2. HOẠT ĐỘNG CM

2.1 HOẠT ĐỘNG CM

Thay đổi cách quản lý

• Theo dõi các yêu cầu thay đổi đối với phần mềm từ khách hàng và nhà phát triển, tính ra chi phí và tác động của các thay đổi, và quyết định các thay đổi nên được thực hiện.

Quản lý phiên bản

• Theo dõi nhiều phiên bản của các thành phần hệ thống và đảm bảo rằng các thay đổi đối với các thành phần do các nhà phát triển khác nhau thực hiện không ảnh hưởng lẫn nhau.

Xây dựng hệ thống

• Quá trình tập hợp các thành phần chương trình, dữ liệu và thư viện, sau đó biên dịch chúng để tạo ra một hệ thống thực thi.

Quản lý phát hành

• Chuẩn bị phần mềm để phát hành bên ngoài và theo dõi các phiên bản hệ thống có

đã được phát hành cho khách hàng sử dụng.

Diagram

Description automatically generated

component version: phiên bản thành phần

system building: xây dựng hệ thống

version management: quản lý phiên bản

system version: phiên bản hệ thống

change proposals: thay đổi đề xuất

change management: thay đổi cách quản lý

system releases: bản phát hành hệ thống

release management: quản lý phát hành

2.2 Multi-version systems - Hệ thống nhiều phiên bản

Đối với các hệ thống lớn, không bao giờ chỉ có một phiên bản "làm việc" của một hệ thống.

• Luôn có một số phiên bản của hệ thống ở các giai đoạn phát triển khác nhau.

• Có thể có một số nhóm tham gia vào việc phát triển các phiên bản hệ thống khác nhau.

2.3 Multi-version system development: Phát triển hệ thống nhiều phiên bản

3 CHANGE MANAGEMENT - THAY ĐỔI CÁCH QUẢN LÝ

3.1 CHANGE MANAGEMENT

• Các nhu cầu và yêu cầu của tổ chức thay đổi trong suốt thời gian tồn tại của hệ thống, các lỗi phải được sửa chữa và các hệ thống phải thích ứng với những thay đổi trong môi trường của chúng.

• Quản lý thay đổi nhằm đảm bảo rằng quá trình phát triển của hệ thống là một quá trình được quản lý và ưu tiên dành cho những thay đổi cấp bách nhất và hiệu quả về chi phí.

• Quá trình quản lý thay đổi liên quan đến việc phân tích chi phí và lợi ích của những thay đổi được đề xuất, phê duyệt những thay đổi đáng giá và theo dõi những thành phần nào trong hệ thống đã được thay đổi.

3.2 The change management process - Quy trình quản lý thay đổi

3.3 A partially completed change request form - Một biểu mẫu yêu cầu thay đổi đã hoàn thành một phần

Thay đổi biểu mẫu yêu cầu

Dự án: SICSA / AppProcessing Number: 23/02

Người yêu cầu thay đổi: I. Sommerville Ngày: 20/07/12

Thay đổi được yêu cầu: Trạng thái của người nộp đơn (bị từ chối, được chấp nhận, v.v.) phải là hiển thị trực quan trong danh sách ứng viên được hiển thị.

Thay đổi máy phân tích: R. Looek Ngày phân tích: 25/07/12

Các thành phần bị ảnh hưởng: ApplicationListDisplay, StatusUpdater

Các thành phần được liên kết: StudentDatabase

Thay đổi biểu mẫu yêu cầu

Đánh giá thay đổi: Tương đối đơn giản để thực hiện bằng cách thay đổi màu hiển thị theo trạng thái. Một bảng phải được thêm vào để liên quan trạng thái với màu sắc. Không cần thay đổi các thành phần liên quan.

Thay đổi mức độ ưu tiên: Trung bình

Thay đổi cách triển khai:

Nỗ lực ước tính: 2 giờ

Ngày đến ứng dụng SGA. tổ: 28/07/12 Ngày quyết định CCB: 30/07/12

Quyết định: Chấp nhận sự thay đổi. Thay đổi sẽ được triển khai trong Bản phát hành 1.2

Người triển khai thay đổi: Ngày thay đổi:

Ngày gửi đến QA: Quyết định của QA:

Ngày gửi cho CM:

Bình luận:

3.4 Factors in change analysis - Các yếu tố trong phân tích thay đổi

• Hậu quả của việc không thực hiện thay đổi

• Lợi ích của sự thay đổi

• Số lượng người dùng bị ảnh hưởng bởi sự thay đổi

• Chi phí thực hiện thay đổi

• Chu kỳ phát hành sản phẩm

4 VERSION MANAGEMENT - QUẢN LÝ PHIÊN BẢN

4.1 Version management - Quản lý phiên bản

• Quản lý phiên bản (VM) là quá trình theo dõi các phiên bản khác nhau của các thành phần phần mềm hoặc các mục cấu hình và hệ thống mà các thành phần này được sử dụng.

• Nó cũng liên quan đến việc đảm bảo rằng các thay đổi do các nhà phát triển khác nhau thực hiện đối với các phiên bản này không ảnh hưởng lẫn nhau.

• Do đó, quản lý phiên bản có thể được coi là quá trình quản lý các đường chuẩn và đường cơ sở.

4.2 Codelines and baselines- Đường mã và đường cơ sở

• •Dòng mã là một chuỗi các phiên bản mã nguồn với các phiên bản mới hơn trong chuỗi bắt nguồn từ các phiên bản trước đó.

• Các dòng mã thường áp dụng cho các thành phần của hệ thống để mỗi thành phần có các phiên bản khác nhau.

• Đường cơ sở là một định nghĩa của một hệ thống cụ thể.

• Do đó, đường cơ sở chỉ định các phiên bản thành phần được bao gồm trong hệ thống cộng với đặc điểm kỹ thuật của các thư viện được sử dụng, tệp cấu hình, v.v

4.3 Baselines - Đường cơ sở

• Các đường cơ sở có thể được chỉ định bằng cách sử dụng

ngôn ngữ cấu hình, cho phép bạn xác định những thành phần nào được bao gồm trong một phiên bản của một hệ thống cụ thể.

• Đường cơ sở rất quan trọng vì bạn thường phải tạo lại một phiên bản cụ thể của một hệ thống hoàn chỉnh.

• Ví dụ, một dòng sản phẩm có thể được khởi tạo để có các phiên bản hệ thống riêng cho các khách hàng khác nhau. Bạn có thể phải tạo lại phiên bản được giao cho một khách hàng cụ thể, ví dụ: nếu khách hàng đó báo cáo lỗi trong hệ thống của họ cần được sửa chữa.

4.4 Codelines and baselines - Đường mã và đường cơ sở

4.5 Version control systems - Hệ thống kiểm soát phiên bản

• Hệ thống kiểm soát phiên bản (VC) xác định, lưu trữ và kiểm soát quyền truy cập vào các phiên bản khác nhau của các thành phần. Có hai loại hệ thống điều khiển phiên bản hiện đại

• Hệ thống tập trung, nơi có một kho lưu trữ chính duy nhất lưu giữ tất cả các phiên bản của các thành phần phần mềm đang được phát triển. Subversion là một ví dụ được sử dụng rộng rãi về hệ thống VC tập trung.

• Hệ thống phân tán, nơi tồn tại nhiều phiên bản của kho thành phần cùng một lúc. Git là một ví dụ được sử dụng rộng rãi về hệ thống VC phân tán.

4.6 Centralized version control - Kiểm soát phiên bản tập trung

- Các nhà phát triển kiểm tra các thành phần hoặc thư mục của các thành phần từ

kho lưu trữ dự án vào không gian làm việc riêng tư của họ và làm việc trên

bản sao trong không gian làm việc riêng tư của họ.

-Khi các thay đổi của họ hoàn tất, họ đăng ký các thành phần trở lại kho lưu trữ.

-Nếu nhiều người đang làm việc trên một thành phần cùng một lúc, thì mỗi người hãy kiểm tra thành phần đó từ kho lưu trữ.

Nếu một thành phần đã được kiểm tra, hệ thống VC sẽ cảnh báo những người dùng khác muốn kiểm tra thành phần đó rằng nó đã được người khác kiểm tra.

4.7 Repository Check-in/Check-out - Đăng ký / Trả phòng kho lưu trữ

4.8 Distributed version control - Kiểm soát phiên bản phân tán

• Kho lưu trữ 'chính' được tạo trên máy chủ duy trì mã do nhóm phát triển tạo ra.

• Thay vì kiểm tra các tệp mà họ cần, một nhà phát triển tạo một bản sao của kho dự án được tải xuống và cài đặt trên máy tính của họ.

• 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 họ.

• Khi các thay đổi được thực hiện, họ ‘cam kết’ những thay đổi này và cập nhật kho lưu trữ máy chủ riêng của họ. Sau đó, họ có thể 'đẩy' những thay đổi này vào kho dự án

4.9 Benefits of distributed version control - Lợi ích của việc kiểm soát phiên bản phân tán

• Nó 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.

• Nó cho phép làm việc ngoại tuyến để các nhà phát triển có thể thực hiện các thay đổi nếu họ không có kết nối mạng.

• Hỗ trợ dự án là cách làm việc mặc định.

Các nhà phát triển có thể biên dịch và kiểm tra toàn bộ hệ thống trên các máy cục bộ của họ và kiểm tra những thay đổi mà họ đã thực hiện.

4.10 Open source development - Phát triển mã nguồn mở

• Kiểm soát phiên bản phân tán là điều cần thiết để phát triển mã nguồn mở.

• Nhiều người có thể làm việc đồng thời trên cùng một hệ thống mà không có sự điều phối trung tâm nào.

• Cũng như một kho lưu trữ riêng trên máy tính của họ, các nhà phát triển cũng duy trì một kho lưu trữ máy chủ công cộng để họ đẩy các phiên bản mới của các thành phần mà họ đã thay đổi.

• Sau đó, ‘người quản lý’ hệ thống nguồn mở quyết định thời điểm đưa những thay đổi này vào hệ thống chính thức.

4.11 Version management systems - Hệ thống quản lý phiên bản

• Nhận dạng phiên bản và phát hành

Các phiên bản được quản lý được chỉ định số nhận dạng khi chúng được gửi lên hệ thống.

• Quản lý lưu trữ

Để giảm không gian lưu trữ theo yêu cầu của nhiều phiên bản thành phần chỉ khác nhau một chút, hệ thống quản lý phiên bản thường cung cấp các phương tiện quản lý lưu trữ.

• Thay đổi lịch sử ghi

Tất cả những thay đổi được thực hiện đối với mã của hệ thống hoặc thành phần đều được ghi lại và liệt kê.

• Phát triển độc lập

Hệ thống quản lý phiên bản theo dõi các thành phần đã được kiểm tra để chỉnh sửa và đảm bảo rằng các thay đổi do các nhà phát triển khác nhau thực hiện đối với một thành phần không gây trở ngại.

• Hỗ trợ dự án

Hệ thống quản lý phiên bản có thể hỗ trợ sự phát triển của một số dự án chia sẻ các thành phần..

4.12 Storage management using deltas - Quản lý bộ nhớ bằng delta

4.13 Branching and merging - Phân nhánh và hợp nhất

• Thay vì một chuỗi phiên bản tuyến tính phản ánh những thay đổi đối với thành phần theo thời gian, có thể có một số chuỗi độc lập.

• Điều này là bình thường trong quá trình phát triển hệ thống, nơi các nhà phát triển khác nhau làm việc độc lập trên các phiên bản khác nhau của mã nguồn và vì vậy hãy thay đổi nó theo những cách khác nhau.

• Ở một số giai đoạn, có thể cần hợp nhất các nhánh dòng mã để

tạo một phiên bản mới của một thành phần bao gồm tất cả các thay đổi đã được thực hiện.

• Nếu các thay đổi được thực hiện liên quan đến các phần khác nhau của mã, các phiên bản thành phần có thể được hợp nhất tự động bằng cách kết hợp các delta áp dụng cho mã.

4.14 Branching and merging - Phân nhánh và hợp nhất

5. SYSTEM BUILDING - XÂY DỰNG HỆ THỐNG

5.1 System building - Xây dựng hệ thống

• Xây dựng hệ thống là quá trình tạo ra một

hệ thống bằng cách biên dịch và liên kết các thành phần hệ thống, thư viện bên ngoài, tệp cấu hình, v.v.

• Các công cụ xây dựng hệ thống và công cụ quản lý phiên bản phải

giao tiếp khi quá trình xây dựng liên quan đến việc kiểm tra các phiên bản thành phần từ kho lưu trữ được quản lý bởi hệ thống quản lý phiên bản.

• Mô tả cấu hình được sử dụng để xác định đường cơ sở cũng được sử dụng bởi công cụ xây dựng hệ thống.

5.2 Build platforms - Xây dựng nền tảng

• Hệ thống phát triển, bao gồm các công cụ phát triển như trình biên dịch, trình chỉnh sửa mã nguồn, v.v.

• Các nhà phát triển kiểm tra mã từ hệ thống quản lý phiên bản vào một không gian làm việc riêng trước khi thực hiện các thay đổi đối với hệ thống.

• Máy chủ xây dựng, được sử dụng để xây dựng các phiên bản hoàn chỉnh, có thể thực thi được của hệ thống.

• Nhà phát triển đăng ký mã vào hệ thống quản lý phiên bản trước khi nó được xây dựng. Việc xây dựng hệ thống có thể dựa vào các thư viện bên ngoài không có trong hệ thống quản lý phiên bản.

• Môi trường đích, là nền tảng mà hệ thống thực thi.

**5.3 Development, build, and target platforms - Nền tảng phát triển, xây dựng và mục tiêu**

**5.4 System building - Xây dựng hệ thống**

5.5 Build system functionality - Xây dựng chức năng hệ thống

• Xây dựng tạo kịch bản

• Tích hợp hệ thống quản lý phiên bản

• Biên dịch lại tối thiểu

• Tạo hệ thống thực thi

• Tự động hóa kiểm tra

• Báo cáo

• Tạo tài liệu

6 RELEASE MANAGEMENT - QUẢN LÝ PHÁT HÀNH

6.1 Release management - Quản lý phát hành

• Bản phát hành hệ thống là một phiên bản của hệ thống phần mềm được phân phối cho khách hàng.

• Đối với phần mềm thị trường đại chúng, thường có thể xác định hai loại bản phát hành: bản phát hành chính cung cấp chức năng mới quan trọng và bản phát hành nhỏ, sửa chữa lỗi và khắc phục sự cố của khách hàng đã được báo cáo.

• Đối với phần mềm tùy chỉnh hoặc các dòng sản phẩm phần mềm, các bản phát hành của hệ thống có thể phải được sản xuất cho từng khách hàng và các khách hàng cá nhân có thể chạy nhiều bản phát hành khác nhau của hệ thống cùng một lúc.

6.2 Release tracking - Theo dõi phát hành

• Trong trường hợp có sự cố, có thể cần phải tạo lại chính xác

phần mềm đã được phân phối cho một khách hàng cụ thể.

• Khi một bản phát hành hệ thống được tạo ra, nó phải được lập thành tài liệu để đảm bảo rằng nó có thể được tạo lại chính xác trong tương lai.

• Điều này đặc biệt quan trọng đối với các hệ thống nhúng tùy chỉnh, có tuổi thọ cao, chẳng hạn như các hệ thống điều khiển các máy móc phức tạp.

Khách hàng có thể sử dụng một bản phát hành duy nhất của các hệ thống này trong nhiều năm và có thể yêu cầu các thay đổi cụ thể đối với một hệ thống phần mềm cụ thể sau ngày phát hành ban đầu.

6.2 Release reproduction - Phát hành tái sản xuất

• Để tạo tài liệu cho một bản phát hành, bạn phải ghi lại các phiên bản cụ thể của các thành phần mã nguồn đã được sử dụng để tạo mã thực thi.

• Bạn phải giữ các bản sao của các tệp mã nguồn, các tệp thực thi tương ứng và tất cả các tệp dữ liệu và cấu hình.

• Bạn cũng nên ghi lại các phiên bản của hệ điều hành, thư viện, trình biên dịch và các công cụ khác được sử dụng để xây dựng phần mềm.

6.3 Release planning - Lập kế hoạch phát hành

• Cũng như các công việc kỹ thuật liên quan đến việc tạo ra một bản phân phối phát hành, tài liệu quảng cáo và công khai phải được chuẩn bị và đưa ra các chiến lược tiếp thị để thuyết phục khách hàng mua bản phát hành mới của hệ thống.

• Thời gian phát hành

Nếu các bản phát hành quá thường xuyên hoặc yêu cầu nâng cấp phần cứng, khách hàng có thể không chuyển sang bản phát hành mới, đặc biệt nếu họ phải trả tiền cho nó.

Nếu các bản phát hành hệ thống quá ít thường xuyên, thị phần có thể bị mất khi khách hàng chuyển sang các hệ thống thay thế.

6.4 Factors influencing system release planning - Các yếu tố ảnh hưởng đến việc lập kế hoạch phát hành hệ thống

|  |  |
| --- | --- |
| Yếu tố | miêu tả |
| Sự cạnh tranh | Đối với phần mềm dành cho thị trường đại chúng, việc phát hành hệ thống mới có thể là cần thiết vì một sản phẩm cạnh tranh đã giới thiệu các tính năng mới và thị phần có thể bị mất nếu chúng không được cung cấp cho khách hàng hiện tại. |
| Yêu cầu tiếp thị | Bộ phận tiếp thị của một tổ chức có thể đã cam kết về việc các bản phát hành sẽ có sẵn vào một ngày cụ thể. |
| Thay đổi nền tảng | Bạn có thể phải tạo bản phát hành mới của ứng dụng phần mềm khi phiên bản mới của nền tảng hệ điều hành được phát hành. |
| Chất lượng kỹ thuật của hệ thống | Nếu các lỗi hệ thống nghiêm trọng được báo cáo ảnh hưởng đến cách nhiều khách hàng sử dụng hệ thống, thì có thể cần phải đưa ra thông cáo sửa chữa lỗi. Các lỗi nhỏ của hệ thống có thể được sửa chữa bằng cách phát hành các bản vá (thường được phân phối qua Internet) có thể được áp dụng cho bản phát hành hiện tại của hệ thống. |

6.5 Release components - Giải phóng các thành phần

Cũng như mã thực thi của hệ thống, một bản phát hành cũng có thể

bao gồm:

• các tệp cấu hình xác định cách thức cấu hình bản phát hành cho các cài đặt cụ thể;

• các tệp dữ liệu, chẳng hạn như tệp thông báo lỗi, cần thiết để hệ thống hoạt động thành công;

• một chương trình cài đặt được sử dụng để giúp cài đặt hệ thống trên phần cứng đích;

• tài liệu điện tử và giấy mô tả hệ thống;

• bao bì và công khai liên quan đã được thiết kế cho bản phát hành đó.

6.6 Key points - Những điểm chính

• Quản lý cấu hình là quản lý một hệ thống phần mềm đang phát triển. Khi duy trì một hệ thống, một nhóm CM được đưa ra để đảm bảo rằng các thay đổi được đưa vào hệ thống một cách có kiểm soát và hồ sơ được duy trì với các chi tiết về những thay đổi đã được thực hiện.

• Các quy trình quản lý cấu hình chính là quản lý thay đổi, quản lý phiên bản, xây dựng hệ thống và quản lý phát hành.

• Quản lý thay đổi liên quan đến việc đánh giá các đề xuất thay đổi từ khách hàng của hệ thống và các bên liên quan khác và quyết định xem việc triển khai những thay đổi này trong phiên bản mới của hệ thống có hiệu quả về chi phí hay không.

• Quản lý phiên bản liên quan đến việc theo dõi các phiên bản khác nhau của các thành phần phần mềm khi chúng được thực hiện thay đổi.

• Xây dựng hệ thống là quá trình lắp ráp các thành phần của hệ thống thành một chương trình thực thi để chạy trên một hệ thống máy tính đích.

• Phần mềm phải thường xuyên được xây dựng lại và kiểm tra ngay sau khi phiên bản mới được xây dựng. Điều này giúp dễ dàng phát hiện các lỗi và sự cố đã được đưa ra kể từ lần xây dựng cuối cùng.

• Bản phát hành hệ thống bao gồm mã thực thi, tệp dữ liệu, tệp cấu hình và tài liệu. Quản lý bản phát hành liên quan đến việc đưa ra quyết định về ngày phát hành hệ thống, chuẩn bị tất cả thông tin để phân phối và ghi lại mỗi bản phát hành hệ thống.