**Bai thuc hanh**

Câu 7. Kể tên 5 CM tool, 3 VC tools (**0.5 điểm**)

Trả lời:

**Configuration management (CM)** là công cụ thực hiện việc thay đổi trạng thái hiện tại của hệ thống sang trạng thái được xác định trước. Hay nói cách khác, là công cụ hỗ trợ, cấu hình, cài đặt hệ thống một cách tự động.

CM tools: Rational ClearCase, Microsoft System Center Configuration Management, Puppet, AnsibleChef, CFEngine, Salt, Vagrant, Bcfg2, SmartFrog, Rudder.

VC Tools:

Trong [kỹ nghệ phần mềm](https://vi.wikipedia.org/wiki/C%C3%B4ng_ngh%E1%BB%87_ph%E1%BA%A7n_m%E1%BB%81m), một **hệ thống quản lý phiên bản** (còn viết là **VCS** theo từ [tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh) *version control systems*) là một hệ thống lưu giữ các phiên bản của [mã nguồn](https://vi.wikipedia.org/wiki/M%C3%A3_ngu%E1%BB%93n) của sản phẩm phần mềm, giúp các [lập trình viên](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_vi%C3%AAn) có thể dễ dàng lấy lại phiên bản mong muốn.

Git (GNU General Public License), Apache Subversion (Apache license), Mercurial (GNU General Public License), Plastic SCM (Proprietary Software), Team foundation server, Perforce (Proprietary Software), Concurrent Version system (GNU General Public License), GNU Bazaar (GNU General Public License), Mirosoft Visual Sourcesafe (Proprietary Software), Rational ClearCaseRevision Control System (GNU General Public License), Dracs (GNU General Public License), PVCS (Proprietary Software), Source code control system (common development and distribution system), Monotone (GNU General Public License), Vault (Proprietary Software), GNU Arch (GNU General Public License), SVK (GNU General Public License), CVSNT (GNU General Public License), CA Harvest Software Change Manager (Proprietary Software), Codeville (BSD licenses), Panvalet (Proprietary Software), Code co-op (Proprietary Software), Bitbucket Server (Proprietary Software), VisualSVN (Proprietary Software), TortoiseCVS (GNU General Public License), KDESvn (GNU General Public License), OpenCVS (BSD licenses), ViewVC (BSD licenses), Rabbit VCS (GNU General Public License), PixelNovel timelineSourceForge Enterprise Edition (Proprietary Software), Adobe Version Cue (Proprietary Software), OpenGrok (common development and distribution system), Cervisia (GNU General Public License), Rubby Version Manager (Apache Licensens).

Câu 5: Những lợi ích mà Configuration Management System mang lại ?

Trả lời:

- Quản lý tập trung các loại cấu hình, các thiết lập, các loại cấu hình một cách có tổ chức & gọn gàng.

- Có thể linh hoạt thay đổi và dễ dàng thay đổi theo nhiều môi trường, nền tảng khác nhau.

- Tránh sự lập đi lặp lại công việc của quản trị viên hệ thống, do đó tránh khỏi thiếu sót, sai xót.

- Triển khai toàn diện hoặc cục bộ cho các máy trạm tùy theo yêu cầu người quản trị hệ thống.

- Tiết kiệm tối đa thời gian, tái sử dụng và tái cơ cấu cấu hình dễ dàng, linh hoạt.

Câu 6: Những lợi ích mà Version Control System mang lại ?

- Có thể quản lý mã nguồn theo dạng tách nhánh (branch) mã nguồn cho mục đích phát triển nhiều chức năng song song nhau mà không sợ ảnh hưởng nhau, khi cần thiết sẽ merge lại tổng thể mã nguồn.

- Giúp làm việc team work dễ dàng hơn, linh hoạt hơn khi phải làm cùng lúc nhiều task, bởi vì bạn có thể cấu trúc công việc dễ dàng hơn.

- Kiểm soát lỗi và nguồn gốc lỗi, vá lỗi trở nên dễ dàng.

- Thỏa sức làm việc ở bất cứ đâu vì chỉ cần clone mã nguồn từ kho chứa hoặc clone một phiên bản thay đổi nào đó từ kho chứa, hoặc một nhánh nào đó từ kho chứa.

- Quản lý được các phiên bản của kho mã nguồn, trích xuất dễ dàng ngay rivision mong muốn bất cứ lúc nào.

Câu 4. CI/CD là gì

CI (Continuos Integration) là phương pháp phát triển phần mềm đòi hỏi các thành viên trong nhóm tích hợp công việc thường xuyên. Mỗi ngày, các thành viên đều phải theo dõi và phát triển công việc của họ ít nhất một lần. Việc này sẽ được một nhóm khác kiểm tra tự động, nhóm này sẽ tiến hành kiểm thử truy hồi để phát hiện lỗi nhanh nhất có thể. Cả nhóm thấy rằng phương pháp tiếp cận này giúp giảm bớt vấn đề về tích hợp hơn và cho phép phát triển phần mềm gắn kết nhanh hơn.

Một số tool CI server: Gitlab CI, Travis CI, Jenkins, Teamcity, Hudson, Team Foundation Server.

<https://oanhnn.github.io/2016-07-08/what-is-continuous-integration.html>

<https://vi.wikipedia.org/wiki/T%C3%ADch_h%E1%BB%A3p_li%C3%AAn_t%E1%BB%A5c>

CD (Continuos Deployment):

Continuous Deployment hay CD là khái niệm có quan hệ rất mật thiết với Continuous Integration. Continuous Deployment giúp việc deploy ứng dụng có thể được diễn ra trong bất cứ thời điểm nào và sau đó thậm chí là tự động release một phiên bản mới trên môi trường test hoặc production nếu như bản build (gồm các mã lệnh được bạn gửi lên server) của bạn vượt qua được tất cả các test case.

Với cách làm như vậy, nếu xảy ra lỗi trong quá trình build (mã lệnh bạn commit không vượt qua các kiểm định về chất lượng hoặc các test case) lập trình viên sẽ có thể dễ dàng xác định được lỗi diễn ra ở đâu và qua đó giúp fix lỗi một cách hiệu quả hơn.

Tham khảo: <https://www.codehub.vn/Continuous-Integration-La-Gi>

Câu 3: Sự khác nhau giữa C Deployment & C Delivery

Tương tự như Continuous Deployment dự án có thể deploy ở bất cứ thời điểm nào. tuy nhiên điều này không có nghĩa là nó có thể đảm bảo ứng dụng có thể được deploy một cách an toàn trên môi trường production vì thông thường sẽ có sự khác biệt trong cách cấu hình ứng dụng giữa môi trường production và test do vậy nó sẽ dừng lại và không thực hiện deploy lên môi trường production.

Câu 2: API là gì **(0.5 điểm**)

Một giao diện lập trình ứng dụng (tiếng Anh Application Programming Interface, viết tắt API) là một [giao diện](https://vi.wikipedia.org/wiki/Giao_di%E1%BB%87n_(khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh)) mà một hệ thống máy tính hay ứng dụng cung cấp để cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, và/hoặc cho phép dữ liệu có thể được trao đổi qua lại giữa chúng. Chẳng hạn, một chương trình máy tính có thể (và thường là phải) dùng các hàm API của [hệ điều hành](https://vi.wikipedia.org/wiki/H%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh) để xin cấp phát bộ nhớ và truy xuất tập tin. Nhiều loại hệ thống và ứng dụng hiện thực API, như các hệ thống đồ họa, cơ sở dữ liệu, mạng, dịch vụ web, và ngay cả một số trò chơi máy tính. Đây là phần mềm hệ thống cung cấp đầy đủ các chức năng và các tài nguyên mà các lập trình viên có thể rút ra từ đó để tạo nên các tính năng giao tiếp người- máy như: các trình đơn kéo xuống, tên lệnh, hộp hội thoại, lệnh bàn phím và các cửa sổ. Một trình ứng dụng có thể sử dụng nó để yêu cầu và thi hành các dịch vụ cấp thấp do hệ điều hành của máy tính thực hiện. Hệ giao tiếp lập trình ứng dụng giúp ích rất nhiều cho người sử dụng vì nó cho phép tiết kiệm được nhiều thời gian tìm hiểu các chương trình mới, do đó khích lệ mọi người dùng nhiều ứng dụng hơn.

<https://vi.wikipedia.org/wiki/Giao_di%E1%BB%87n_l%E1%BA%ADp_tr%C3%ACnh_%E1%BB%A9ng_d%E1%BB%A5ng>

Câu 1: MVC là gì ? (**0.5 điểm**)

MVC (viết tắt của Model-View-Controller) là một mẫu [kiến trúc phần mềm](https://vi.wikipedia.org/wiki/Ki%E1%BA%BFn_tr%C3%BAc_ph%E1%BA%A7n_m%E1%BB%81m) để tạo lập [giao diện người dùng](https://vi.wikipedia.org/wiki/Giao_di%E1%BB%87n_ng%C6%B0%E1%BB%9Di_d%C3%B9ng) trên máy tính. MVC chia một ứng dụng thành ba phần tương tác được với nhau để tách biệt giữa cách thức mà thông tin được xử lý nội hàm và phần thông tin được trình bày và tiếp nhận từ phía người dùng.

Khi sử dụng đúng cách, mẫu MVC giúp cho người phát triển phần mềm cô lập các nguyên tắc nghiệp vụ và giao diện người dùng một cách rõ ràng hơn. Phần mềm phát triển theo mẫu MVC tạo nhiều thuận lợi cho việc bảo trì vì các nguyên tắc nghề nghiệp và giao diện ít liên quan với nhau.

Trong mẫu Model-View-Controller, mô hình (model) tượng trưng cho [dữ liệu](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u) của chương trình phần mềm. Tầm nhìn hay khung nhìn (view) bao gồm các thành phần của giao diện người dùng. Bộ kiểm tra hay bộ điều chỉnh (controller) quản lý sự trao đổi giữa dữ liệu và các nguyên tắc nghề nghiệp trong các thao tác liên quan đến mô hình.

<https://vi.wikipedia.org/wiki/MVC>