# Câu 2: Chất lượng phần mềm là gì? Có thể đo chất lượng phần mềm không? Đo bằng cách nào? Các thước đo chất lượng phần mềm?

***Chất lượng phần mềm định nghĩa theo IEEE là:***

* Mức độ mà một hệ thống, thành phần hoặc một tiến trình đạt được yêu cầu đã đặc tả
* Mức độ mà một hệ thống, thành phần hoặc một tiến trình đạt được những nhu cầu hay mong đợi của khách hàng hoặc người sử dụng.

# Câu 12: SQA và mục tiêu của SQA? Các khía cạnh trong SQA? Yêu cầu chung của SQA? Các cách tiếp cận trong SQA?

***Đảm bảo chất lượng phần mềm***

Là một tập các hoạt động đã được lập kế hoạch và có hệ thống, cần thiết để cung cấp đầy đủ sự tin cậy vào quy trình phát triển phần mềm hay quy trình bảo trì phần mềm của sản phẩm hệ thống phần mềm phù hợp với các yêu cầu chức năng kỹ thuật cũng như với các yêu cầu quản lý mà giữ cho lịch biểu và hoạt động trong phạm vi ngân sách.

***Mục tiêu:***

Sản xuất ra phần mềm có chất lượng cao thỏa mãn khách hàng trong khuôn khổ thời gian, ngân sách và chất lượng.

***Các khía cạnh trong SQA:***

* *Kế hoạch ĐBCL*
* Mô tả chất lượng mong muốn, thiết lập các tiêu chuẩn chất lượng và cách đánh giá (đo) các thuộc tính chất lượng.
* Định rõ qui trình đánh giá chất lượng.
* Định rõ các chuẩn mực về quản lí (dùng chuẩn có sẵn/thiếp lập mới).
* *Kiểm soát chất lượng*

Bao gồm chuỗi các hoạt động: thanh tra, kiểm duyệt, kiểm thử để đảm bảo sản phẩm tuân theo đặc tả.

* *Đảm bảo chất lượng*

Xác nhận và báo cáo về qui trình để cung cấp thông tin quản lí và ra quyết định.

***Yêu cầu chung của SQA:***

* Tuân thủ đặc tả là nền tảng để đo lường chất lượng.
* Các chuẩn được xác định trước dùng để phát triển các tiêu chí chất lượng và dẫn dắt quá trình kỹ nghệ.
* Bên cạnh tuân thủ các yêu cầu tường minh (trong đặc tả), phần mềm phải tuân thủ các đặc tả không tường minh như dễ dùng, dễ bảo trì, tin cậy.

***Các cách tiếp cận trong SQA:***

1. Chứng minh đúng đắn (logic Hoare).
2. Thống kê chất lượng
   * Thông tin về hỏng hóc được thu thập và phân loại
   * Xác định nguyên nhân hỏng hóc
   * Áp dụng nguyên lý Pareto (80% các hỏng hóc có thể được tìm thấy trong 20% **modules chương trình)** để cô lập nguyên nhân hỏng hóc.
3. Cleanroom: tổ hợp hai điểm trên
   * Phát triển theo mô hình tăng trưởng
   * Đặc tả hình thức
   * Kiểm tra tĩnh bằng cách dùng các lí lẽ đúng đắn
   * Kiểm tra động (testing) để xác nhận độ tin cậy.
   * Ngăn ngừa hỏng hóc hơn là loại bỏ lỗi

# Câu 13: Làm thế nào để ĐBCL? Hệ thống đảm bảo chất lượng: Mục tiêu, các thành phần?

***Làm thế nào để ĐBCL:***

Nguyên tắc 1 : bài bản

* Qui trình đảm bảo chất lượng
  + Chỉ rõ cách thức tiến hành ĐBCL
  + Cách thức kiểm tra, giám sát ĐBCL
* Có tài liệu, số liệu về công tác ĐBCL: minh chứng
  + Tài liệu về mọi hoạt động trong qui trình pm
  + Tài liệu, số liệu kiểm tra giám sát
  + Tài liệu đánh giá chất lượng: kế hoạch, số liệu

Nguyên tắc 2: không ngừng cải tiến

* Kế hoạch
* Thực hiện
* Kiểm tra
* Cải tiến

***Hệ thống đảm bảo chất lượng: Mục tiêu, các thành phần***

Mục tiêu:

* Tối thiểu hóa số lỗi phần mềm
* Đạt được mức chất lượng đòi hỏi
  + 1. thành phần trong hệ thống ĐBCL:
       1. Tiền dự án
* Kế hoạch về thời gian và ngân sách phải được thiết lập một cách phù hợp
* Kế hoạch phát triển phần mềm & đảm bảo chất lượng phải được xác định rõ.
  + - 1. Đánh giá các hoạt động trong vòng đời dự án

Hai giai đoạn :

* Phát triển
* Kiểm tra – thẩm tra - xác nhận
* Xét duyệt
* Ý kiến chuyên gia
* Kiểm thử
* Vận hành, bảo trì
  + - 1. Thành phần ngăn chặn lỗi và cải tiến

1. Thành phần quản lí chất lượng phần mềm
2. Thành phần chuẩn hóa, chứng nhận và đánh giá hệ thống ĐBCL

* Thiết lập chuẩn quốc tế và chuyên nghiệp trong quản lí tổ chức.
* Các chuẩn cho qui trình phần mềm: tập trung vào các hướng dẫn (”how”) cho đội ngũ phát triển
* Các chuẩn quản lí chất lượng: tập trung vào cái gì (what) được đòi hỏi cho một hệ thống quản lí chất lượng

1. Thành phần tổ chức nhân sự cho ĐBCL

Tổ chức & phát triển đội ngũ SQA(người quản lí, đội kiểm thử, đội SQA,…)

# Câu 14: Các yếu tố chất lượng của McCall

Theo McCall, các yếu tố chất lượng phần mềm được chia làm ba loại :

(1) Vận hành sản phẩm: tính chính xác, tin cậy, hiệu quả, tính toàn vẹn, sử dụng được

(2) Xem xét lại sản phẩm: tính bảo trì, linh hoạt, có thể test được

(3) Chuyển giao sản phẩm: tính khả chuyển, có khả năng giao tác, có khả năng sử dụng lại.

# Câu 9: Công việc QA (Quality assurance) và QC (Quality Control) giống và khác nhau như thế nào? Trong một tổ chức có cần phải có đủ cả nhân viên QA và QC không hay chỉ cần một trong hai? Giải thích? Để làm tốt công việc QA, QC cần có những yêu cầu gì?

***Công việc QA (Quality assurance) và QC (Quality Control) giống và khác nhau như thế nào?***

***Giống:***

Cả hai bộ phận này đều có mục đích cuối cùng là tạo ra được sản phẩm chất lượng cao, thỏa mãn khách hàng.

Là 2 công việc đều nhằm đảm bảo chất lượng phần mềm

***Khác:***

|  |  |
| --- | --- |
| **QA** | **QC** |
| **QA** đưa ra quy trình làm việc giữa các bên liên quan để đảm bảo chất lượng sản phẩm | **QC** thực hiện công việc kiểm tra chất lượng sản phẩm. |
| Mục đích của QA là ngăn chặn trước lỗi | Mục đích của QC là xác định và cải thiện lỗi |
| Công việc của **QA** đó là quản lý, định hướng chất lượng | **QC** là xác thực chất lượng |
| **QA** không tham gia đến việc thực thi chương trình | **QC** tham gia thực thi chương trình để khắc phục lỗi. |
| **QA** đảm bảo bạn đang làm đúng | **QC** đảm bảo kết quả của những gì bạn đã làm đúng với kết quả bạn mong muốn |
| **QA** đưa ra chuẩn mực trong kế hoạch, phương pháp thực hiện của dự án. | **QC** đảm bảo rằng những chuẩn mực, quy tắc đã đề ra sẽ được áp dụng trong sản phẩm. |
| QA là quá trình để tạo ra các sản phẩm | QC là quá trình xác minh sản phẩm đã hoàn thiện |

***Trong một tổ chức có cần phải có đủ cả nhân viên QA và QC không hay chỉ cần một trong hai? Giải thích?***

Trong một tổ chức cần phải có đủ cả nhân viên QA và QC .

Vì:

QA là bộ phận bao quát, chịu trách nhiệm tổng thể về tiêu chuẩn, quy trình để đảm bảo chất lượng. QC là bộ phận thực hiện những quy định, hướng dẫn của QA, kiểm tra chất lượng cụ thể của sản phẩm hoàn thiện hay các công đoạn trong sản xuất.

VD QC và QA sẽ tương tác với nhau như sau:

* QA đưa ra quy trình làm việc cho team phát triển sản phẩm. Trong đó có khâu Testing, quy định QC kiểm thử sản phẩm ở giai đoạn nào, sử dụng công cụ gì, tiêu chuẩn nào là sản phẩm đạt yêu cầu.
* QC thực thi quy trình mà QA đề ra.
* QA giám sát, theo dõi và kiểm tra QC có thực hiện đúng quy trình không. Sau đó chỉnh sửa cho phù hợp với tiến độ, hiện trạng dự án.

***Để làm tốt công việc QA, QC cần có những yêu cầu gì?***

Để làm tốt công việc QA cần có những yêu cầu sau:

* Hiểu sâu về kiến trúc hệ thống của phần mềm
* Khả năng tổ chức, tư duy logic và có hệ thống
* Kỹ năng phân tích, làm việc dựa trên số liệu tốt
* Kiến thức rộng về các lĩnh vực phần mềm mà các team đang thực hiện
* Kỹ năng giao tiếp trong nội bộ team và các team khác

Mục đích khai thác thông tin về sản phẩm, dự án và ứng dụng nó vào việc xây dựng hệ thống quy trình.

* Hiểu rõ về các chứng chỉ CMMI, ISO,… trong phần mềm để xây dựng các quy trình chuẩn cho các team
* Tính tỉ mỉ, lòng kiên nhẫn và biết thừa nhận lỗi sai

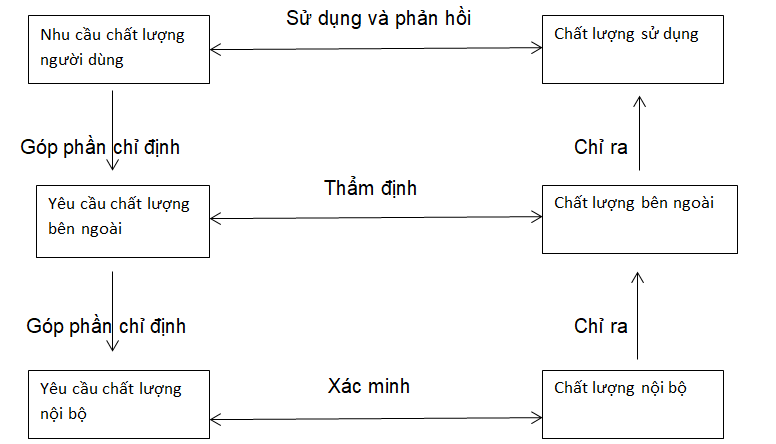
Để làm tốt công việc QC cần có những yêu cầu sau:

* Kỹ năng code (nếu làm Automation)
* Có kiến thức tốt về mọi chức năng, khía cạnh của sản phẩm. Như vậy mới xét duyệt được các yêu cầu
* Người làm QC cần cẩn thận, kỹ tính. Vì công việc này đòi hỏi sự chính xác cao độ và chú ý đến mọi khía cạnh có thể có vấn đề của sản phẩm.
* Kỹ năng giao tiếp tốt. Vì QC phải làm việc với rất nhiều thành viên khác trong team và nhất là công việc truyền đạt ý kiến, phản hồi của mình với Developer, Project Manager.

# Câu 5: Đội ngũ đảm bảo chất lượng phần mềm là những ai?

* Bộ phận QA: bao quát, chịu trách nhiệm tổng thể về tiêu chuẩn, quy trình để đảm bảo chất lượng.
* Bộ phận QC: thực hiện những quy định, hướng dẫn của QA kiểm tra chất lượng cụ thể của sản phẩm hoàn thiện hoàn thiện hay các công đoạn trong sản xuất.

# Câu 16: Trình bày khung chất lượng



Theo sơ đồ này ta đi từ trên xuống dưới và từ trái qua phải.

Nhu cầu chất lượng người dùng góp phần xác định yêu cầu chất lượng bên ngoài từ đó xác định yêu cầu chất lượng nội bộ

Từ yêu cầu chất lượng nội bộ xác minh chất lượng nội bộ

Chất lượng nội bộ chỉ ra chất lượng bên ngoài đồng thời chất lượng ngoài được thẩm định lại với yêu cầu chất lượng ngoài trước đó.

Chất lượng bên ngoài tác động đến chất lượng người dùng. Quá trình lại được lặp lại qua sử dụng và phản hồi.

# Câu 6: Đảm bảo chất lượng phần mềm có giống như kiểm thử phần mềm không? Tại sao?

Đảm bảo chất lượng phần mềm

* DBCL bao quát, chịu trách nhiệm tổng thể về tiêu chuẩn, quy trình để đảm bảo chất lượng
* Trọng tâm của đảm bảo chất lượng phần mềm là kiểm soát chất lượng và đáp ứng các yêu cầu. Đảm bảo chất lượng phần mềm được định hướng theo quy trình nhằm ngăn ngừa lỗi xảy ra.
* Mục đích chính là đảm bảo chất lượng của phần mềm sẽ được phân phối.

Kiểm thử phần mềm

* Mặt khác, Kiểm thử phần mềm chỉ đơn giản là quá trình tìm lỗi trong phần mềm bằng các trường hợp kiểm thử.

# Câu 11: Những yếu tố bên ngoài là gì? Và những yếu tố này đóng góp vào việc đảm bảo chất lượng dự án phần mềm như thế nào? Mục tiêu đảm bảo chất lượng từ sự đóng góp của các yếu tố bên ngoài là gì? Có các công cụ đảm bảo chất lượng của các thành viên đóng góp bên ngoài không? Liệt kê?

***Những yếu tố bên ngoài là gì?***

Những người tham gia bên ngoài có thể được phân chia thành 3 nhóm chính:

* Subcontractors( những nhà thầu phụ, hiện nay được gọi là những tổ chức “outsourcing”)
* Những nhà cung cấp COTS software và những module phần mềm sử dụng lại.
* Khách hàng

***Mục tiêu đảm bảo chất lượng từ sự đóng góp của các yếu tố bên ngoài:***

* Để tránh trì hoãn hoàn thành nhiệm vụ và để đảm bảo cảnh báo sớm tính trước sự trì hoãn.
* Để đảm bảo mức độ chất lượng có thể chấp nhận được của bộ phận triển khai và đón nhận cảnh báo sớm của phạm vi chất lượng yêu cầu.
* Để đảm bảo đủ tài liệu phục vụ cho nhóm bảo trì
* Để đảm bảo liên tục, toàn diện và đáng tin cậy kiểm soát việc thực hiện người tham gia bên ngoài.

***Có các công cụ đảm bảo chất lượng của các thành viên đóng góp bên ngoài không? Liệt kê?***

* xem xét lại tài liệu yêu cầu.
* Đánh giá các tiêu chuẩn chọn lựa liên quan đến các thành viên đóng góp bên ngoài.
* Thành lập ủy ban điều khiển gia nhập và kết hợp của dự án.
* Sự đóng góp trong sự xem xét thiết kế.
* Sự đóng góp trong kiểm tra phần mềm.
* Cách trình bày các thủ tục đặc biệt
* Xác định các team leader của các nhà cung cấp và các thành viên.
* Chuẩn bị các báo cáo tiến trình phát triển của các hoạt động phát triển dự án.
* Xem xét lại các tài liệu và acceptance tests

# Câu 10: Vòng đời dự án CNTT là gì? Tích hợp các hoạt động chất lượng trong vòng đời dự án là thế nào?

***Tích hợp các hoạt động chất lượng trong vòng đời dự án***

Các mô hình phát triển phần mềm cung cấp các khái niệm và các phương pháp luận cần thiết để xây dựng phần mềm. Mô hình phát triển bao gồm các định nghĩa các hoạt động chính cần cho sự phát triển của dự án, cho các điểm mốc của dự án.

Người quản trị dự án sẽ xác định cách dự án thực hiện bằng việc xác định mô hình phát triển và áp dụng vào trong dự án.

Hầu hết các hoạt động đảm bảo chất lượng được đặt trong cấu hình với các điểm mốc, trong đó yêu cầu xem xét lại các hoạt động phát triển sản phẩm đã được hoàn thiện trước đó. Do đó, những người đảm bảo chất lượng phần mềm nên hiểu biết về các mô hình kỹ nghệ phần mềm khác nhau để có thể chuẩn bị một kế hoạch chất lượng mà sẽ được tích hợp thành kế hoạch dự án.

# Câu 15: Một số độ đo cơ bản để đảm bảo chất lượng phần mềm

***KLOC***

KLOC - thước đo cổ điển này đo kích cỡ của phần mềm bởi hàng ngàn dòng lệnh.

Việc áp dụng độ đo KLOC phải sử dụng cùng một ngôn ngữ lập trình hay công cụ phát triển.

Độ đo này chỉ có thể chính xác sau khi dự án đã kết thúc.Tuy nhiên bằng kinh nghiệm, hoặc bằng thống kê tương tự có thể ước lượng được khối lượng mã nguồn của một phần mền trước khi kết thúc dự án.

KLOC sau khi kết thúc dự án sẽ được dùng để ước lượng các dự án tương tự sau này.

***Độ đo điểm chức năng( funtion point metric)***

Đo các tài nguyên phát triển (nguồn lực con người) được yêu cầu để phát triển một chương trình, dựa trên chức năng đã xác định cho hệ thống phần mềm.

**FP = Điểm của các yếu tố chính x (0.65 + 0.01 x ΣFi)**

Các yếu tố chính bao gồm:

* số lượng người dùng đầu vào
* số lượng đầu ra của người dùng
* số lượng người dùng yêu cầu
* số lượng tập tin
* Số lượng các giao tiếp ngoài

Mỗi Fi được từ 0 đến 5 điểm tùy theo mức độ

FP định lượng phần mềm bằng số lượng các chức năng mà hệ thống cung cấp cho người sử dụng chủ yếu dựa trên thiết kế logic. FP có thể sử dụng để ước lượng quy mô ứng dụng phần mềm. Quy mô là thành phần quan trọng trong việc xác định hiệu suất (đầu vào, đầu ra), dự tính nhân lực, nắm bắt chi phí , ...

***Độ đo FPA***

FPA là một phương pháp định lượng phần mềm dựa trên các chức năng nghiệp vụ của hệ thống khi bàn giao sản phẩm cho khách hàng. FPA phân tích hệ thống thành các chức năng nhỏ, dễ hiểu.

Một số công thức tính

Năng suất lao động: Năng suất = kết quả đầu ra/đầu vào

Năng suất đầu vào = FP/đầu vào

*Tính toán Function Point Count(FPC)*

FPC là tích của thời gian VAF và UAF

FPC = VAF.UAF

FPC: function point count

VAF: giá trị hiệu chỉnh

UAF: giá trị mặc định

Công thức với mục đính tính toán ra số lượng chức năng phát sinh trong dự án.

***Độ đo bảo trì phần mềm***

**SMI = [MT - (Fc + Fa + Fd)] / MT**

Chỉ số trưởng thành phần mềm (SMI), được chuẩn hóa bởi IEEE, cung cấp một dấu hiệu về tính ổn định của sản phẩm phần mềm dựa trên những thay đổi xảy ra cho mỗi lần phát hành sản phẩm. Thông tin cơ bản cần thiết để tính SMI là:

* Mt = Số lượng mô-đun trong bản phát hành hiện tại
* Fc = Số lượng mô-đun trong bản phát hành hiện tại đã được thay đổi
* Fa = Số lượng mô-đun trong bản phát hành hiện tại đã được thêm
* Fd = Số lượng mô-đun từ bản phát hành trước đã bị xóa trong bản phát hành hiện tại

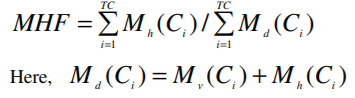
Khi SMI tiến đến 1.0, sản phẩm bắt đầu ổn định. SMI có thể được sử dụng như một thước đo để lập kế hoạch cho các hoạt động bảo trì phần mềm.

Ngoài ra : sự hài lòng của khách hàng, mật độ lỗi, nhận diên thương hiệu

***MOOD(độ đo thiết kế HDT)***

hệ thống hướng đối tượng ngày càng trở nên phổ biến, nên các kỹ sư phần mềm phải có các phép đo định lượng hoặc truy cập vào chất lượng của các thiết kế ở cả cấp độ kiến trúc và thành phần.

***MHF: tỉ lệ phương thức ẩn trong hệ thống***



Mh (Ci) = số lượng phương thức ẩn trong Ci

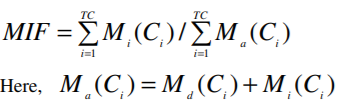
Md (Ci) = số lượng phương thức được định nghĩa trong lớp Ci

Mv (Ci) = số lượng phương thức hiển thị trong lớp Ci

TC được định nghĩa là tổng số lớp.

* Nếu giá trị của MHF cao (100%), điều đó có nghĩa là tất cả các phương thức là riêng tư chỉ ra rất ít chức năng. Do đó, không thể sử dụng lại các phương pháp có MHF cao.
* MHF với giá trị thấp (0%) cho thấy tất cả các phương thức đều công khai, điều đó có nghĩa là hầu hết các phương thức đều không được bảo vệ.

***MIF: Tỉ lệ phương thức kế thừa trong hệ thống***



Mi (Ci) = số lượng phương thức được kế thừa trong Ci

Ma (Ci) = số lượng phương thức khả dụng được định nghĩa trong lớp Ci

Md (Ci) = số lượng phương thức được định nghĩa trong lớp Ci

* Nếu giá trị của MIF thấp (0%), điều đó có nghĩa là không có phương thức nào tồn tại trong lớp như cũng như lớp thiếu một tuyên bố thừa kế.

***POF: Tỉ lệ đa hình của hệ thống.***



M0 (Ci)= số lượng phương thức ghi đè trong lớp Ci

Mn (Ci) = số lượng phương thức mới được định nghĩa trong lớp Ci

DC (Ci)= con cháu trong Ci

* Giá trị của POF có thể thay đổi trong khoảng từ 0% đến 100%.
* Nếu một dự án có 0% POF, nó chỉ ra rằng dự án không sử dụng đa hình
* 100% POF chỉ ra rằng tất cả các phương thức được ghi đè trong tất cả các lớp dẫn xuất.

***COF: Tỉ lệ ghép nối của hệ thống***

**COF = Khớp nối thực tế / Khớp nối tối đa có thể**

* 0% COF cho thấy không có lớp nào được ghép nối
* 100% COF cho thấy tất cả các lớp được ghép với tất cả các lớp khác. Giá trị cao COF nên tránh.

# Câu 8: Cần lưu ý những thành phần chất lượng phần mềm tiền dự án nào?

Rà soát hợp đồng, kế hoạch phát triển dự án phần mềm và kế hoạch chất lượng phần mềm.

Rà soát hợp đồng: xem xét lại những bản dự thảo của những tài liệu đề xuất và hợp đồng.

Kế hoạch phát triển dự án phần mềm: lập thời gian biểu phát triển, ước lượng được nguồn nhân lực, yêu cầu và ngân sách, xác định mô hình kỹ nghệ, xác định rủi ro phát triển.

Kế hoạch chất lượng phần mềm: xác định các tiêu chuẩn quy trình đảm bảo chất lượng phần mềm, quy trình đánh giá chất lượng.

# Câu 7: Việc xây dựng hợp đồng thực hiện dự án phần mềm có quan trọng không? Có cần thiết phải rà soát hợp đồng trước khi thực hiện không?

Rất quan trọng ! Đã từ lâu pháp luật về hợp đồng chiếm một vị trí rất quan trọng trong hệ thống pháp luật Việt Nam. Mục đích là nhằm bảo vệ quyền tự do ý chí của các bên. Một hợp đồng tồi – thường mô tả các yêu cầu không chặt chẽ và đưa ra kế hoạch cũng như ngân sách phi thực tế =>dẫn đến một phần mềm có chất lượng tồi.

Cần thiết phải rà soát hợp đồng trước khi thực hiện để đảm bảo các điều khoản phù hợp cho các bên liên quan, cải thiện ngân sách và thời gian biểu, là những cơ sở cho những đề nghị và hợp đồng sau này, đồng thời có thể biết được những rủi ro tiềm tàng sớm (trong mục tiêu ban đầu và trong bản dự thảo hợp đồng).

# Câu 17: Việc sử dụng các chuẩn trong CNPM mang lại những lợi ích gì?

* Giúp đảm bảo chất lượng cao (bằng cách check các yêu cầu của chuẩn)
* Chỉ ra được chất lượng sản phẩm.
* Gia tăng tính đồng bộ giữa các nhóm và đào tạo nhân viên mới
* Cung cấp hướng dẫn thực hành tốt/đúng- tránh sai lầm trong quá khứ
* Cung cấp sự bảo hộ về mặt luật pháp.
* Chuẩn làm cho các sản phẩm, dịch vụ có nguồn gốc khác nhau có thể so sánh được hay nối kết được.

# Câu 19: Nội dung ISO 9126 trong CNPM

**ISO 9126** là [tiêu chuẩn quốc tế](https://vi.wikipedia.org/w/index.php?title=Ti%C3%AAu_chu%E1%BA%A9n_qu%E1%BB%91c_t%E1%BA%BF&action=edit&redlink=1) đánh giá [phần mềm](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m).

3 góc nhìn về chất lượng SP phần mềm:

* + ***Chất lượng trong:*** là các đặc tính của SP phần mềm từ góc nhìn bên trong trong suốt quá trình phát triển hoặc bảo trì (ví dụ code, kiến trúc).
  + ***Chất lượng ngoài:*** là các đặc tính của SP phần mềm từ góc nhìn bên ngoài thông qua thực thi chương trình.
  + ***Chất lượng dùng:*** là góc nhìn của người dùng về chất lượng SP phần mềm khi dùng nó trong một môi trường, hoàn cảnh cụ thể. Nó phản ánh mức độ mà người dùng có thể hoàn thành mục tiêu của họ trong một hoàn cảnh cụ thể hơn là đánh giá các đặc tính của bản thân phần mềm

⇒Chất lượng trong xác định chất lượng ngoài, chất lượng ngoài ảnh hưởng đến chất lượng dùng

Chất lượng trong và chất lượng ngoài gồm 6 đặc trưng :

1. **Chức năng PM:**

Khả năng của phần mềm cung cấp các chức năng tương ứng với yêu cầu đã được phát biểu và suy diễn khi được dùng trong điều kiện cụ thể.Gồm : tính thích hợp, tính chính xác,khả năng tương tác, tính an toàn

1. **Tin cậy**

Khả năng phần mềm đạt được một mức độ hiệu quả xác định khi dùng trong điều kiện cụ thể.Gồm : tính chín chắn, khả năng chịu lỗi, khả năng phục hồi

1. **Khả dụng**

Khả năng phần mềm liên quan tới tính dễ hiểu, dễ học, hấp dẫn khi được dùng trong một điều kiện cụ thể. Gồm : hiểu được, học được, vận hành được, tính hấp dẫn

* Chú ý: trong trường hợp ứng dụng web, tính dùng được còn bao gồm cả truy cập được.

1. **Hiệu quả**

khả năng phần mềm cung cấp hiệu năng thích hợp nhằm tiết kiệm tối đa tài nguyên và tăng tối đa hiệu suất công viêc trong điều kiện cụ thể.Gồm : thời gian xử lý, sử dụng tài nguyên.

1. **Bảo trì được**

Khả năng của phần mềm cho phép sửa đổi, nâng cấp, bao gồm sửa chữa, cải tiến hoặc thích nghi của phần mềm thay đổi cho phù hợp với môi trường, các yêu cầu và chức năng mới. Gồm : khả năng phân tích, khả năng thay đổi được, khả năng kiểm thử được, tính ổn định.

1. **Khả chuyển**

Khả năng phần mềm chuyển từ môi trường này sang môi trường khác. Gồm : khả năng thích nghi, khả năng cài đặt, khả năng chung sống, khả năng thay thế được.

**Chất lượng dùng**

* Tính hiệu quả: khả năng phần mềm cho phép người dùng đạt được mục đích một cách chính xác và hoàn toàn trong điều kiện cụ thể
* Tính năng suất: khả năng phần mềm cho phép người dùng sử dụng lượng tài nguyên hợp lý tương đối để thu được hiệu quả công viêc trong điều kiện cụ thể
* Tính an toàn: phần mềm có thể đáp ứng mức độ rủi ro chấp nhận được đối với người sử dụng trong điều kiện cụ thể.
* Tính thỏa mãn: phần mềm có khả năng làm thỏa mãn người sử dụng trong điều kiện cụ thể.

# Câu 20: Phiên bản phần mềm là gì? Trình bày sự đóng góp của quản lý cấu hình phần mềm trong việc đảm bảo chất lượng phần mềm.

***Trình bày sự đóng góp của quản lý cấu hình phần mềm trong việc đảm bảo chất lượng phần mềm.***

Quản lý cấu hình đóng vai trò quan trọng trong đảm bảo chất lượng phần mềm:

* Một lỗi nào đó của phần mềm đang xây dựng đã tốn nhiều công sức sửa chữa, bỗng “thình lình” xuất hiện trở lại.
* Một chức năng nào đó của phần mềm đã được phát triển và kiểm tra cẩn thận bổng thất lạc hoặc biến mất một cách khó hiểu.
* Một chương trình đã được kiểm tra hết sức cẩn thận, bỗng nhiên không “chạy” được nữa.
* Một chương trình gồm nhiều đơn thể (module), mỗi đơn thể gồm nhiều chức năng, các chức năng được chia ra cho nhiều lập trình viên, mỗi chức năng bao gồm nhiều tập tin mã nguồn với nhiều phiên bản (version) khác nhau Khi tích hợp hệ thống và biên dịch, trong hàng chục tập tin mã nguồn với hàng trăm version, tập tin nào, version nào là đúng và cần phải lấy để tiến hành tích hợp?

Các vấn đề trên sẽ không xảy ra nếu như trong dự án, việc QLCH được thực hiện nghiêm túc và kiểm soát chặt chẽ.

QLCH tốt sẽ giải quyết được hàng loạt những khó khăn

• Cập nhật đồng thời: Khi 2 hoặc nhiều lập trình viên làm việc cách biệt nhau nhưng trên cùng một chương trình hoặc dự án, những thay đổi mà người này thực hiện có thể sẽ phá vỡ kết quả làm việc của người khác

• Chia sẻ source code: Trong các hệ thống lớn, khi các chức năng chung bị thay đổi, tất cả những người liên quan phải được biết.

# Câu 1: Phần mềm là gì? Quy trình phát triển phần mềm? Các đặc trưng của phần mềm?

Là hệ thống bao gồm những thành phần sau đây:

* Chương trình máy tính
* Các thủ tục
* Tài liệu liên quan
* Dữ liệu cần thiết cho sự vận hành của hệ thống

***Quy trình phát triển phần mềm:***

Là một tập hợp các hoạt động tổ chức mà mục đích của chúng là xây dựng và phát triển phần mềm.Có một số [mô hình](https://vi.wikipedia.org/wiki/Quy_tr%C3%ACnh_ph%C3%A1t_tri%E1%BB%83n_ph%E1%BA%A7n_m%E1%BB%81m#M%C3%B4_h%C3%ACnh_ph%C3%A1t_tri%E1%BB%83n_ph%E1%BA%A7n_m%E1%BB%81m) cho việc xây dựng các quy trình này, mỗi mô hình mô tả các phương thức cũng như [các nhiệm vụ hoặc thao tác](https://vi.wikipedia.org/w/index.php?title=C%C3%A1c_giai_%C4%91o%E1%BA%A1n_c%E1%BB%A7a_v%C3%B2ng_%C4%91%E1%BB%9Di_ph%C3%A1t_tri%E1%BB%83n_ph%E1%BA%A7n_m%E1%BB%81m&action=edit&redlink=1) cần được thực hiện trong cả quá trình.

***Các đặc trưng của phần mềm:***

* Phần mềm được kỹ nghệ, không được chế tạo theo nghĩa cổ điển
* Phần mềm không mòn cũ, hỏng hóc, nhưng thoái hóa
* Phần lớn phần mềm vẫn được xây dựng theo đơn đặt hàng.
* Sự phức tạp và tính thay đổi luôn là bản chất
* Ngày nay phần mềm được phát triển theo nhóm
* Phần mềm là vô hình

***Chất lượng phần mềm là có thể đo được***

***Đo bằng cách nào?***

Có thể đo trực tiếp và đo gián tiếp.

**Độ đo trực tiếp** là độ đo có thể tính đếm trực tiếp không thông qua các độ đo khác (ví dụ độ đo LOC – lines of code), có **độ đo gián tiếp** là các độ đo tính qua các độ đo khác (ví dụ tỉ lệ lỗi = số lỗi / số dòng mã nguồn).

Để thực hiện đo chất lượng phần mềm cần:

* Xác định các thuộc tính cần đo
* Xác định các độ đo chất lượng phần mềm để đo thuộc tính
* Phân tích thống kê dữ liệu độ đo đã thu thập được.
* Các thay đổi cần thiết để cải thiện chất lượng
* Tiến hành đo lại và thực hiện những bước trên.

***Các thước đo chất lượng phần mềm?***

* Theo giai đoạn của phần mềm
  + Các độ đo liên quan tới qui trình phát triển: Kích thước hệ thống (KLOC), đo mật độ lỗi, Function Point, độ đo thiết kế, độ đo hình thái hệ thống,…
  + Các độ đo liên quan tới qui trình vận hành và bảo trì: độ đo bảo trì, độ đo hài lòng khách hàng, FPA,…
* Theo chủ đề được đo
  + Chất lượng
  + Thời gian
  + Hiệu quả (trong loại bỏ lỗi hoặc dùng tài nguyên)

# Câu 4: Phần mềm có số dòng lệnh nhiều hay ít thì chất lượng hơn?

Tùy từng trường hợp mà phần mềm có số dòng lệnh nhiều hơn hay ít hơn thì tốt hơn.

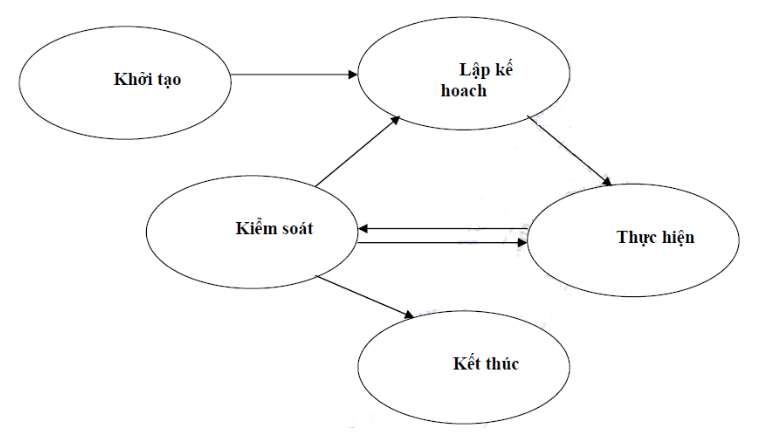
Phần mềm có số dòng lệnh ít thì các biến sử dụng ít =>tốn ít bộ nhớ đồng thời thể hiện được tính chuyên nghiệp tối ưu nâng cao hiệu suất. Tuy nhiên code quá ngắn dẫn đến khó bảo trì cho người phía sau.

Dự án là một thực thể thống nhất, thời gian thực hiện xác định và có độ bất định nhất định nên các tổ chức, đơn vị thường chia dự án thành một số giai đoạn để quản lý thực hiện. Mỗi gian đoạn được đánh dấu bằng việc thực hiện một hay nhiều công việc. Tổng hợp các giai đoạn này được gọi là vòng đời của dự án.

Vòng đời của dự án xác định thời điểm bắt đầu, thời điểm kết thúc và thời gian thực hiện dự án.

Vòng đời dự án xác định những công việc nào sẽ được thực hiện trong từng giai đoạn và ai sẽ tham gia thực hiện.

Các giai đoạn thường có cơ chế tự hoàn thiện thông qua các công việc giám sát, đánh giá.



Tiến trình phát triển PM tổng quát:

1. Khảo sát (xác định y/c, đặc tả)
2. Phân tích
3. Thiết kế
4. Code
5. Kiểm thử
6. Chuyển giao (bàn giao cho KH)
7. Bảo trì: 50%, 30%, 40%, 20%

# Câu 18: Sự khác nhau giữa CMM/CMMi

* CMM đến trước nhưng sau đó được cải tiến và đã thành công bởi CMMI.
* Ban đầu, CMM mô tả cụ thể về công nghệ phần mềm trong khi CMMI mô tả quy trình và quy trình được tích hợp vì nó áp dụng cho cả phần mềm và kỹ thuật hệ thống.
* CMMI hữu ích hơn và phổ quát hơn CMM cũ.
* CMMi cũng bao gồm 5 mức như CMM: khởi đầu, lặp lại được, được định nghĩa, được quản lý và tối ưu.

Tuy nhiên Một điểm đặc biệt là mỗi doanh nghiệp có thể áp dụng mô hình CMM ở bất kỳ mức nào mà không cần tuân theo bất kỳ một qui định nào không cần phải đạt mức thấp trước rồi mới có thể đạt mức cao còn CMMI thì ngược lại level đều tuân theo một chuẩn ở mức độ cao hơn. Muốn đạt được chuẩn cao hơn thì các chuẩn của các level trước phải thoả mãn.

***Phiên bản phần mềm là gì?***

Phiên bản (version): là một thể hiện của phần mềm mà có sự thay đổi so với các thể hiện khác của phần mềm đó

* Thay đổi có thể bao gồm: chức năng mới, cải tiến hiệu năng, sửa đổi lỗi của phiên bản cũ… Một số phiên bản có thể tương đương hoàn toàn về mặt chức năng nhưng được thiết kế để hoạt động cho các cấu hình phần mềm hay phần cứng khác nhau