**Báo cáo đảm bảo chất lượng phần mềm**

**Nhóm 9**

**Trả lời 10 câu hỏi**

**Câu 1:** **Lỗi phần mềm là gì? Nguyên nhân gây ra lỗi phần mềm?**

-**Lỗi phần mềm - Software Error**: Là các phần code sai do lỗi cú pháp, logic hoặc lỗi do phân tích, thiết kế.

**- Nguyên nhân gây ra lỗi:**  
1. Lỗi khi định nghĩa yêu cầu  
2. Quan hệ Client-developer tồi  
3. Sai phạm có chủ ý với yêu cầu phần mềm  
4. Lỗi thiết kế logic  
5. Lỗi lập trình  
6. Không tuân thủ các hướng dẫn viết tài liệu và code  
7. Thiếu sót của quá trình kiểm thử  
8. Lỗi giao diện người dùng và thủ tục  
9. Lỗi tài liệu

**Câu 2: Kể ra các độ đo đặc trưng chất lượng chính của McCall? Giải thích nội dung của nó?**

McCall có 11 tiêu chí; chia thành chia thành 3 nhóm:

- Tiêu chí vận hành sản phẩm:

+ Tính đúng đắn – Correctness: Đặc tả về độ chính xác, tính toàn vẹn, thời gian của outputs.

+ Tính tin cậy Reliability Requirements: Yêu cầu về tính tin cậy đề cập tới lỗi khi cung cấp dịch vụ – Định ra tỷ lệ lỗi cho toàn hệ thống hoặc cho các chức năng mong muốn.

+ Tính hiệu quả - Efficiency Requirements: Đề cập tới tài nguyên phần cứng cần để thực hiện các chức năng của phần mềm.

+ Tính toàn vẹn – Integrity: đề cập tới bảo mật hệ thống với việc ngăn chặn truy cập trái phép

+ Tính khả dụng – Usability Requirements: đề cập tới quy mô của nguồn lực cần để đào tạo nhân viên mới và để sử dụng hệ thống phần mềm.

- Tiêu chí sửa đổi sản phẩm:

+ Tính bảo trì được - Maintainability Requirements: Mức công sức cần đề xác định nguyên nhân (tìm ra vấn đề) cho hỏng hóc phần mềm và để sửa hỏng hóc đó, và để xác minh việc sửa thành công. ; Đề cập tới cấu trúc môđun của phần mềm, tài liệu chương trình bên trong, tài liệu của lập trình viên, kiến trúc và chi tiết thiết kế và tài liệu tương ứng.

+ Tính linh hoạt – Flexibility Requirements: đề cập tới nguồn lực để thay đổi phần mềm cho loại khách hàng khác (mong muốn sử dụng phần mềm với khác biệt nhỏ);

+ Tính kiểm thử được - Testability Requirements: Có lưu lại kết quả trung gian để hỗ trợ test? Có tạo file log, backup?

- Tiêu chí chuyển giao sản phẩm:

+ Khả năng di động-Portability Requirements: nếu phần mềm phải cài trong môi trường mới (phần cứng khác, hệ điều hành khác,…) và vẫn duy trì môi trường cũ, tính di động là cần thiết

+ Khả năng tái sử dụng - Reusability Requirements: Ta có thể tái sử dụng các phần của phần mềm cho ứng dụng khác? – Có thể tiết kiệm nhiều chi phí phát triển cho tìm lỗi / kiểm thử. –Chắc chắn thu được phần mềm chất lượng cao và phát triển nhanh hơn –Dùng nhiều

+ Tính tương thích-Interoperability Requirements: phần mềm có cần giao tiếp với các hệ thống đã có

**Câu 3:** **Trình bày kỹ thuật Walkthrough**

Kỹ thuật Walkthrough là kĩ thuật đánh giá không chính thức, vì vậy sẽ không có người quản lý và giám đốc dự án tham gia. Những người tham gia phải xem tài liệu trước cuộc họp vài ngày, tác giả giải thích tài liệu cho nhóm. Thành viên nhóm tham gia là : tác giả, điều phối viên, giám định viên, đại diện người dùng, chuyên gia bảo trì.

Mọi người sẽ đặt câu hỏi hoặc cho ý kiến bổ sung về một số lĩnh vự để đảm bảo chất lượng kĩ thuật của tài liệu (hoặc sản phẩm)

Buổi giám định có thể xảy ra bất kì lúc nào, bấy kì đâu trong các giai đoạn phát triển phần mềm, mục đích chỉ là tìm lỗi nhanh không tìm giải pháp. Sau giám định tác giả là người phải sửa lỗi

**Câu 4: Inspection**

Là kỹ thuật đánh giá chính thức, tài liệu, sản phẩm… được những người khôn gphair là tác giả hoặc trực tiếp liên quan kiểm tra một cách chi tiết để phát hiện lỗi, các vi phạm tiêu chuẩn, hoặc các vấn đề khác nếu có

* Về cơ bản, nó được tổ chức và thực hiện chặt chẽ hơn walkthrough, vai trò của những người tham gia được phân định rõ ràng. Tài liệu chuẩn bị cho việc xem xét được chuẩn bị chu đáo
* Quá trình duyệt thảo bắt đầu sang giai đoạn code và unit test. Sau buổi họp các lỗi sẽ được sửa lại,rồi đem ra duyệt thảo lại cho đến khi đạt tiêu chuẩn mới kết thúc quá trình này.

**Câu 5: Mục tiêu của rà soát (review)? Liệt kê các loạt rà soát**

**\*Mục tiêu của rà soát (review):**

*Mục tiêu trực tiếp:*   
 a. Để phát hiện lỗi phân tích và thiết kế cũng như các subject  
cần sửa sai, chỉnh sửa và hoàn thiện.  
 b. Để phát hiện các rủi ro mới có khả năng ảnh hưởng tới dự  
án  
 c. Để định vị các sai lệch so với mẫu, hình thức procedures và  
quy ước (templates, style procedures and conventions)  
 d. Để phê duyệt sản phẩm phân tích thiết kế. Việc phê duyệt  
cho phép nhóm tiếp túc pha phát triển tiếp theo.  
*Mục tiêu gián tiếp:*  
 a. Để cung cấp một điểm gặp không chính thức cho trao đổi  
kiến thức chuyên môn về phương pháp, công cụ và kỹ  
thuật.  
 b. Để ghi lại lỗi phân tích, thiết kế mà sẽ dùng như cơ sở cho  
các hoạt động sửa chữa tương lai

**\*Các loại rà soát cần thiết:**

-Web Test Review

-Test Plan Review

-SRS Review

-GUI Review

-Code Review

-Database Design Review

-Test Case Review

-System Test Review

-Proposal Review

-Design Review

**Câu 6: Đảm bảo chất lượng phần mềm là gì? Thường gồm những hoạt động nào?**

Đảm bảo chất lượng phần mềm là gì? Thường gồm những hoạt động nào?

Có rất nhiều định nghĩa về chất lượng phần mềm được đưa ra bởi các tổ chức, cá nhân khác nhau. Trong phạm vi của bài viết này trình bày một số định nghĩa tiêu biểu.

\***Định nghĩa theo IEEE(1991):**

* **Định nghĩa 1:** Chất lượng phần mềm là một mức độ mà một hệ thống, thành phần hệ thống hay tiến trình đáp ứng được yêu cầu đã được đặc tả.
* **Định nghĩa 2:** Chất lượng phần mềm là mức độ mà một hệ thống, thành phần hệ thống hay tiến trình đáp ứng được yêu cầu và sự mong đợi của khách hàng hay người sử dụng.

\***Định nghĩa theo Pressman:** Chất lượng phần mềm là sự phù hợp của các yêu cầu cụ thể về hiệu năng và chức năng, các tiêu chuẩn phát triển phần mềm được ghi lại rõ ràng bằng tài liệu với các đặc tính ngầm định của tất cả các phần mềm được phát triển chuyên nghiệp.

Định nghĩa của Pressman đề xuất ba yêu cầu với chất lượng phần mềm phải được đáp ứng khi phát triển phần mềm:

* Các yêu cầu chức năng rõ ràng là nhân tố chính quyết định chất lượng đầu ra của phần mềm.
* Các tiêu chuẩn chất lượng phần mềm sẽ được nói đến trong hợp đồng.
* Các đặc tính ngầm định cần được đáp ứng trong quá trình phát triển cho dù không được nói đến rõ ràng trong hợp đồng.

**Định nghĩa đảm bảo chất lượng phần mềm**

**Định nghĩa theo Daniel Galin**: Đảm bảo chất lượng phần mềm (Software Quality Assure) là một tập hợp các hành động cần thiết được lên kế hoạch một cách hệ thống để cung cấp đầy đủ niềm tin rằng quá trình phát triển phần mềm phù hợp để thành lập các yêu cầu chức năng kỹ thuật cũng như các yêu cầu quản lý theo lịch trình và hoạt động trong giới hạn ngân sách.

**Câu 7: Mục tiêu của SQA là gì? Các hoạt động chính đảm bảo chất lượng phần mềm là những hoạt động nào?**

SQA là một cách hệ thống, tập hợp các hành động được lập kế hoạch cần thiết để đem lại sự tin cậy là tiến trình phát triển phần mềm/tiến trình bảo trì tuân thủ theo các yêu cầu kỹ thuật về chức năng đã đưa ra cũng như yêu cầu quản lý về giữ lịch trình và hoạt động trong phạm vi ngân sách

Mục tiêu trong phát triển phần mềm:

1. Đảm bảo một mức độ tin cậy chấp nhận được là phần mềm sẽ tuân thủ các yêu cầu kỹ thuật về chức năng.
2. Đảm bảo một mức độ tin cậy chấp nhận được là phần mềm sẽ tuân thủ các yêu cầu quản lý về thời gian và tài chính.
3. Khởi đầu và quản lý các hoạt động để phát triển phần mềm và các hoạt động SQA được cải thiện và đạt hiệu quả cao hơn.

Mục tiêu trong bảo trì phần mềm:

1. Đảm bảo một mức độ tin cậy chấp nhận được là các hoạt động bảo trì phần mềm sẽ tuân thủ yêu cầu kĩ thuật về chức năng
2. Đảm bảo một mức độ tin cậy được là các hoạt động bảo trì phần mềm sẽ tuân theo các yêu cầu quản lý về thời gian và tài chính
3. Khởi đầu và quản lý các hoạt động để bảo trì phần mềm được cải tiến và hiệu quả

7 hoạt động chính:

1. Áp dụng công nghệ kĩ thuật hiệu quả (phương pháp, công cụ)
2. Tiến hành rà soát kỹ thuật chính thức
3. Thực hiện kiểm thử nhiều tầng
4. Tuân theo các chuẩn phát triển
5. Kiểm soát tài liệu phần mềm và sự thay đổi của chúng
6. Thực hiện đo lường
7. Báo cáo và quản lý các báo cáo

**Câu 8: Mô hình Mccall**

Theo mô hình mccall, các yếu tố chất lượng phần mềm được chia ra làm 3 loại

1. Các yếu tố hoạt động của sản phẩm bao gồm tính chính xác, tin cậy, hiệu quả, tính toàn vẹn, sử dụng được

ví dụ:

1. Tính đúng đắn - Correctness.

-Đặc tả về độ chính xác của outputs, ví dụ NLT <1% lỗi,  
có thể gây ra bởi dữ liệu không chính xác hoặc lỗi tính toán;  
- Đặc tả về sự toàn vẹn của outputs, có thể chịu ảnh hưởng  
bởi dữ liệu không đầy đủ  
- Đặc tả về thời gian nhận outputs (thời gian giữa xử lý  
của phần mềm và sự kiện tương ứng)  
- Xác định chuẩn cho coding và viết tài liệu phần mềm

1. Tính tin cậy

–Hệ thống giám sát trung tâm phải có tỷ lệ lỗi nhỏ hơn 1 phần triệu.  
–Thời gian chết -Downtime của một hệ thống phải thấp hơn 10 phút/tháng  
–MTBF và MTTR – cũ và máy móc nhưng vẫn được áp dụng.

1. Tính hiệu quả

–Ta thường dùng MIPS, MHz (vòng/giây); khả năng lưu trữ dữ liệu đo bằng MBhoặc TB; đường truyền (đo bằng KBPS, MBPS, hoặc GBPS)

1. Sử dụng được

Ví dụ spec: Một nhân viên phải có thể xử lý n giao dịch/1 đơn vịthời gian

1. Các yếu tố rà soát bao gồm tính bảo trì, linh hoạt, có thể test được

Ví dụ

1. Tính bảo trì

kích thước của mô đun <= 30 statements.

1. Các yếu tố chuyển giao bao gồm tính khả chuyển, có khả năng sử dụng lại, có khả năng giao tác.

**Câu 9: Giải thích sự khác nhau giữa validation và verification.**

- Verification(xác minh)– tiến trình đánh giá một system hoặc component xem sản

phẩm của một pha phát triển đã cho có thoả mãn điều kiện đưa ra ở đầu pha không.

(Kiểm tra điều có đang làm đúng hay không, ví dụ kiểm tra xây dựng sản phẩm đúng

quy trình không.)

- Validation(xác nhận)– tiến trình đánh giá một system hoặc component trong hoặc sau

development process để xác định xem nó có thảo mãn yêu cầu đã đặc tả hay không.

(Kiểm tra thực hiện có theo hướng đúng không, chẳng hạn kiểm tra thực hiện phần

mềm theo đúng yêu cầu khách hàng không.)

- **Khác nhau** :

* +Verification là tĩnh (Static) trong khi Validation là động (Dynamic).

VD: Verification phần mềm là kiểm thử từng dòng mã, từng hàm. Với Validation, chạy

phần mềm và tìm lỗi. Vị trí lỗi có thể tìm thấy với Verification, mà không thể với Validation.

* + Verification: thẩm tra quan tâm đến việc ngăn chặn lỗi giữa các công đoạn. Phát

hiện lỗi lập trình.

* + Validation: xác nhận quan tâm đến sản phẩm cuối cùng không còn lỗi. Phát hiện lỗi

phân tích, thiết kế.

**Câu 10: Giải thích sự khác nhau giữa failure, error, và fault.**

- Lỗi phần mềm (Software Error) là các phần code sai do lôi cú pháp, logic hoăc lỗi do

phân tích, thiết kế.

+ Thường là chỉ một lỗi của con người trong quá trình xây dựng phần mềm.

- Sai sót (Software Fault) là các errors dẫn tới hoạt động không chính xác của phần mềm.

Không phải error nào cũng gây ra fault.

+ Là lỗi nằm trong mã nguồn, tài liệu của chương trình. Loại lỗi này có nhiều nguyên

nhân như: do error của con người, do công nghệ phức tạp, áp lực công việc, do các

thành phần của hệ thống tương tác với nhau, … Kiểm thử viên chủ yếu là bắt các loại lỗi

này.

- Hỏng (Software Failures) Fault sẽ trở thành failure khi nó được kích hoạt. Một số đường

chạy gây ra failures, một số không.

+ Dùng để chỉ các lỗi dưới góc độ của hệ thống. Khi một hệ thống không thực hiện

được chức năng cần thiết, hoặc thực hiện chức năng không được phép làm thì được gọi

là fail/failure. Một bug có thể là nguyên nhân của nhiều fail khi hệ thống hoạt động.