

NGÂN HÀNG CÂU HỎI

HỌC PHẦN: CÔNG NGHỆ PHẦN MỀM

- Câu 1: Trình bày khái niệm sản phẩm phần mềm. Các nhóm sản phẩm phần mềm hiện có? Phân tích các tiêu chuẩn của một sản phẩm phần mềm và giải thích lý do?.....5
- Câu 2: Trong quá trình xây dựng một phần mềm, vì sao nhóm phát triển dự án cần phải tuân thủ và thực hiện theo mô hình phát triển phần mềm? Cho biết sự khác biệt giữa phần mềm phát triển đại trà và phần mềm phát triển theo yêu cầu.5
- Câu 3: Khủng hoảng phần mềm là gì? Phân tích mối quan hệ giữa khủng hoảng kinh tế và khủng hoảng phần mềm. Bài học thực tế về khủng hoảng phần mềm giai đoạn 2000-2015 của thế giới?.....6
- Câu 4: Liệt kê các pha phát triển trong vòng đời phần mềm. Vì sao pha bảo trì là pha tốn kém thời gian và kinh phí nhất?.....7
- Câu 5: Liệt kê những khó khăn trong phát triển phần mềm. Theo bạn, khó khăn nào ảnh hưởng tới chất lượng phần mềm nhất? Vì sao?.....7
- Câu 6. *Phân tích vòng đời phát triển phần mềm*.....8
- Câu 7. *Mô tả lịch sử phát triển của công nghệ phần mềm. trình bày các tiêu chuẩn để có 1 phần mềm tốt*.....10
- Câu 8. trình bày tóm tắt các loại hồ sơ (tài liệu) đi kèm sản phẩm phần mềm khi bàn giao với người sử dụng.....11
- Câu 9. trình bày đặc trưng cơ bản của mô hình gia tăng. so sánh ưu nhược điểm của mô hình gia tăng và mô hình bản mẫu nhanh.....12
- Câu 10. trình bày đặc trưng cơ bản của mô hình hợp nhất. mô hình này thường ứng dụng trong dự án phần mềm nào.....13
- Câu 11: Dựa trên phân tích cụ thể, hãy chỉ ra những ưu điểm của bản mẫu nhanh trong tiến trình phát triển phần mềm.....14

- Câu 12: Trình bày các đặc trưng cơ bản của mô hình gia tăng và mô hình xoắn ốc. So sánh ưu nhược điểm của mô hình gia tăng và mô hình xoắn ốc.....14
- Câu 13: Mô hình xoắn ốc được cho là phù hợp trong phát triển các dự án phần mềm lớn. Hãy giải thích dựa trên đặc điểm của mô hình này.....16
- Câu 14: Trình bày mô hình thác nước và mô hình xoắn ốc. Nêu ưu và nhược điểm và sự khác nhau của hai mô hình đó?.....16
- Câu 15: Trình bày các đặc trưng cơ bản của mô hình làm bản mẫu nhanh. So sánh ưu nhược điểm của mô hình làm bản mẫu nhanh và mô hình thác nước.. .18
- Câu 16: Trình bày các đặc trưng cơ bản của mô hình thác nước và mô hình chữ V. So sánh ưu nhược điểm của mô hình mô hình thác nước và mô hình chữ V.19
- Câu 18 : Mô hình hợp nhất được cho là phù hợp trong phát triển các dự án phần mềm lớn. Hãy giải thích dựa trên đặc điểm của mô hình này.....21
- Câu 19 : Dựa trên phân tích cụ thể, hãy chỉ ra những ưu điểm của mô hình bản mẫu nhanh trong tiến trình phát triển phần mềm.....22
- Câu 20 : . Trình bày các đặc trưng cơ bản của mô hình làm bản mẫu nhanh. So sánh ưu nhược điểm của mô hình làm bản mẫu nhanh và mô hình thác nước.. .22
- Câu 21: Dựa trên phân tích cụ thể, hãy chỉ ra những ưu điểm của mô hình xoắn ốc trong tiến trình phát triển phần mềm.....22
- Câu22: Phân tích các hoạt động chính trong giai đoạn phân tích phần mềm. Tại sao việc xác định nhu cầu người sử dụng và phân tích tính khả thi là những bước đầu tiên trong quá trình phân tích phần mềm?.....23
- Câu 23: Trình bày nguyên tắc thiết kế giao diện người dùng? Theo bạn nguyên tắc nào quan trọng nhất? Vì sao?.....26
- Câu 24: Khái niệm kiểm thử. So sánh ưu nhược điểm của phương pháp kiểm thử trên bàn (kiểm thử tĩnh) và kiểm thử trên máy (kiểm thử động).....28
- Câu 25: Các lỗi có thể mắc phải trong quá trình cài đặt phần mềm là gì? Nguyên nhân? Theo bạn lỗi nào ảnh hưởng tới việc triển khai phần mềm nhất? Vì sao?29
- Câu 26: Chất lượng của thiết kế được thể hiện qua các yếu tố gì? Các yếu tố đó có vai trò như thế nào trong thiết kế phần mềm?.....29

- Câu 27: Trình bày các giai đoạn để thiết kế một sản phẩm phần mềm? Liệt kê các phương pháp thiết kế phần mềm hiện nay mà bạn biết?.....30
- Câu 28: Nêu các nguyên tắc thiết kế giao diện người dùng ? Theo bạn nguyên tắc nào quan trọng nhất? Vì sao?.....31
- Câu 29: Trình bày phương pháp thiết kế kiểm thử hộp đen (What) và kiểm thử hộp trắng (How).....32
- Câu 30: Liệt kê các phương pháp phân tích thiết kế phần mềm. Trình bày mối quan hệ giữa phân tích thiết kế và đặc tả phần mềm?.....33
- Câu 31. Liệt kê các công việc cần thực hiện trong kỹ thuật kiểm thử hộp đen? Nó giúp kiểm tra những nội dung nào của đối tượng kiểm thử? Ưu điểm của kỹ thuật kiểm thử hộp đen là gì?.....34
- Câu 32. Thiết kế ca kiểm thử trong kỹ thuật kiểm thử hộp trắng cần dựa trên cơ sở nào? Thiết kế ca kiểm thử cần đảm bảo điều kiện gì?.....35
- Câu 33. Nêu các nội dung chính trong giai đoạn đặc tả phần mềm ? Theo bạn những nội dung nào ảnh hưởng tới chất lượng phần mềm?.....35
- Câu 34. Tại sao nói tính linh hoạt và khách quan là yêu cầu quan trọng trong quá trình phân tích và xác định yêu cầu phần mềm.....36
- Câu 35. Các lỗi có thể mắc phải trong quá trình thiết kế phần mềm là gì? Nguyên nhân? Theo bạn lỗi nào là lỗi nặng nhất? Vì sao?.....36
- Câu 36. Trình bày các giai đoạn để thiết kế một sản phẩm phần mềm? Liệt kê các phương pháp thiết kế phần mềm hiện nay mà bạn biết?.....39
- Quá trình thiết kế.....39
- Câu 37. Thế nào là một thiết kế tốt? Chất lượng của thiết kế được thể hiện qua các yếu tố gì? Các yếu tố đó có vai trò như thế nào trong thiết kế phần mềm?..40
- Câu 38. Định nghĩa kiểm thử trong phát triển phần mềm, trình bày những điểm cần lưu ý khi thực hiện kiểm thử.....44
- Câu 39: So sánh hai phương pháp: kiểm thử trên bàn (kiểm thử tĩnh) và kiểm thử trên máy (kiểm thử động).....44

Câu 40: Khái niệm ca kiểm thử? Mục tiêu thiết kế ca kiểm thử? các bước để xây dựng một ca kiểm thử?.....	44
Đề 1:.....	53
Câu 1: sử phát triển của phần mềm: gồm 3 giai đoạn.....	53
Câu 3: hệ thống thông tin quản lý cửa hàng bán linh kiện máy tính: khi khách hàng yêu cầu mua hàng.....	54
1.nên sử dụng mô hình chữ V để xây dựng phần mềm.....	55
2. yêu cầu chức năng và phi chức năng.....	55
3. hệ thống gồm những thực thể và thuộc tính.....	56
đề 2:.....	57
câu 1:khái niệm vòng đời phần mềm.....	57
Câu 2: nêu các nguyên tắc thiết kế giao diện.....	58
Câu 3 : Hệ thống quản lý bãi xe.....	59
1. yêu cầu chức năng và phi chức năng, miền ứng dụng.....	60

Câu 1: Trình bày khái niệm sản phẩm phần mềm. Các nhóm sản phẩm phần mềm hiện có? Phân tích các tiêu chuẩn của một sản phẩm phần mềm và giải thích lý do?

- Sản phẩm phần mềm là gì: là một tập hợp gồm những câu lệnh hoặc chỉ thị được viết bằng một hoặc nhiều ngôn ngữ lập trình, theo một trật tự xác định, và các dữ liệu hay tài liệu liên quan nhằm thực hiện một số nhiệm vụ hay chức năng hoặc giải quyết một vấn đề cụ thể nào đó.

- Các tiêu chuẩn của sản phẩm phần mềm:

+ Tính đúng đắn: Phần mềm phải thực hiện đầy đủ và chính xác các yêu cầu của người dùng

+ Tính tiến hóa: Cho phép người dùng có thể khai báo các thay đổi về qui định của phần mềm theo các thay đổi trong thế giới thực VD: - Cách tính trung bình kiểm tra, trung bình môn - Cách tính lương dựa vào bảng chấm công - Công thức tính tiền phạt - Công thức tính tiền điện

+ Tính tiện dụng: Dễ học, dễ sử dụng, có giao diện trực quan, tự nhiên

+ Tính hiệu quả: Sử dụng tối ưu các tài nguyên (CPU, bộ nhớ, ...) --> sử dụng được trên cấu hình tối thiểu

+ Tính tương thích: Trao đổi dữ liệu được với các phần mềm khác có liên quan VD: - Nhập danh mục sách từ tập tin Excel - Xuất danh sách điểm thành tập tin Excel - Gửi báo cáo tổng kết thông qua Email bằng phần mềm WinFax.

Câu 2: Trong quá trình xây dựng một phần mềm, vì sao nhóm phát triển dự án cần phải tuân thủ và thực hiện theo mô hình phát triển phần mềm? Cho biết sự khác biệt giữa phần mềm phát triển đại trà và phần mềm phát triển theo yêu cầu.

- Tại sao phải tuân thủ và thực hiện theo mô hình phát triển:

Cũng như mọi ngành sản xuất khác, qui trình là một trong những yếu tố cực kỳ quan trọng đem lại sự thành công cho các nhà sản xuất phần mềm, nó giúp cho mọi thành viên trong dự án từ người cũ đến người mới, trong hay ngoài công ty đều có thể xử lý đồng bộ công việc tương ứng vị trí của mình thông qua cách thức chung của công ty, hay ít nhất ở cấp độ dự án.

Mô hình phát triển phần mềm là một thể hiện trừu tượng của quy trình phần mềm. Tùy theo mô hình phát triển phần mềm, các nhóm công việc được triển khai theo những

cách khác nhau. Để sản xuất cùng một sản phẩm phần mềm người ta có thể dùng các mô hình khác nhau. Tuy nhiên không phải tất cả các mô hình đều thích hợp cho mọi ứng dụng. Chính vì vậy, nhóm phát triển dự án sẽ phải chọn ra mô hình thích hợp nhất, tuân thủ và thực hiện theo mô hình đó.

- Sự khác biệt giữa phần mềm đại trà và phần mềm làm theo yêu cầu riêng:
Phần mềm ĐẠI TRÀ là phần mềm làm cho 1 lĩnh vực nào đó thật cụ thể (ví dụ có một số phần mềm chỉ viết riêng để quản lý GAS có trên thị trường, họ ko viết cho lĩnh vực khác mà chỉ đi vào 1 thị trường đó mà thôi), hoặc có tính dùng chung rất cao (ví dụ như phần mềm kế toán) để có thể bán với 1 số lượng cực lớn mà ko phụ thuộc vào sự đặc thù nào của doanh nghiệp. Chi phí phát triển PM đại trà được phân bổ cho hàng chục nghìn khách hàng nên giá các sản phẩm phần mềm đại trà thường khá rẻ. Ngoài ra, phần mềm đại trà còn có tính liên tục phát triển, các phiên bản sau sẽ có chất lượng cao hơn phiên bản trước để đáp ứng được nhu cầu người dùng.

Phần mềm PHÁT TRIỂN THEO YÊU CẦU RIÊNG: Nhà phát triển phần mềm sẽ phải xây dựng dự án để phát triển riêng theo yêu cầu của từng KH. Loại phần mềm này có đặc điểm là chi phí phát triển phần mềm rất lớn và chỉ được phân bổ cho 1 hoặc 1 vài khách hàng riêng biệt nên thường có giá cao. Khi kết thúc hợp đồng thì thường rất ít khi phần mềm được phát triển tiếp trừ phi: 1 là KH ký hợp đồng nạp cấp phần mềm theo yêu cầu; 2 là: có KH mới tương tự có thể áp dụng (nhưng số lượng mới sẽ ko nhiều tới mức như thị trường của phần mềm đại trà).

Câu 3: Khủng hoảng phần mềm là gì? Phân tích mối quan hệ giữa khủng hoảng kinh tế và khủng hoảng phần mềm. Bài học thực tế về khủng hoảng phần mềm giai đoạn 2000-2015 của thế giới?

- Khủng hoảng phần mềm là gì :
10/1968 tại Hội nghị của NATO các chuyên gia phần mềm đã đưa ra thuật ngữ “Khủng hoảng phần mềm” (Software crisis). Qua hàng chục năm, thuật ngữ này vẫn được dùng và ngày càng mang tính cấp bách.

Khủng hoảng là gì ?

+ Điểm ngoặt trong tiến trình của bất kỳ cái gì; thời điểm, giai đoạn hoặc biến cố quyết định hay chủ chốt

+ Điểm ngoặt trong quá trình diễn biến bệnh khi trở nên rõ ràng bệnh nhân sẽ sống hay chết

Trong phần mềm: Là sự day dứt kinh niên (kéo dài theo thời gian hoặc thường tái diễn, liên tục không kết thúc) gặp phải trong phát triển phần mềm máy tính, như:

+ Phải làm thế nào với việc giảm chất lượng vì những lỗi tiềm tàng có trong phần mềm?

+ Phải giải quyết thế nào khi thiếu kỹ thuật viên phần mềm?

+ Phải chế tác phần mềm ra sao khi có yêu cầu phát triển theo qui cách mới xuất hiện ?

+ Phải xử lý ra sao khi sự cố phần mềm gây ra những vấn đề xã hội ?

+ Phần mềm càng lớn sẽ kéo theo phức tạp hóa và tăng chi phí phát triển

2 ý sau không biết làm

Câu 4: Liệt kê các pha phát triển trong vòng đời phần mềm. Vì sao pha bảo trì là pha tốn kém thời gian và kinh phí nhất?

- Liệt kê các pha:

Vòng đời phần mềm là thời kì tính từ khi phần mềm được tạo ra cho tới khi chết đi (từ lúc hình thành đáp ứng yêu cầu, vận hành, bảo dưỡng cho đến khi loại bỏ không dùng nữa).

Vòng đời phần mềm được phân chia thành các pha chính: khảo sát dự án, phân tích, thiết kế, chế tạo, kiểm thử, bảo trì.

- Tại sao bảo trì là pha tốn kém thời gian và kinh phí nhất:

Bảo trì phần mềm được định nghĩa là việc sửa đổi một phần mềm sau khi đã bàn giao để chỉnh lại các lỗi phát sinh, cải thiện hiệu năng của phần mềm hoặc các thuộc tính khác, hoặc làm cho phần mềm thích ứng trong một môi trường đã bị thay đổi. Bảo trì phần mềm có thể chiếm đến 65%-75% công sức trong chu kỳ sống của một phần mềm.

Câu 5: Liệt kê những khó khăn trong phát triển phần mềm. Theo bạn, khó khăn nào ảnh hưởng tới chất lượng phần mềm nhất? Vì sao?

- Các khó khăn trong phát triển phần mềm:

(1) Không có phương pháp mô tả rõ ràng định nghĩa yêu cầu của người dùng (khách hàng), sau khi bàn giao

sản phẩm dễ phát sinh những trục trặc (troubles)

- (2) Với những phần mềm quy mô lớn, tư liệu đặc tả đã cố định thời gian dài, do vậy khó đáp ứng nhu cầu thay đổi của người dùng một cách kịp thời trong thời gian đó
- (3) Nếu không có Phương pháp luận thiết kế nhất quán mà thiết kế theo cách riêng (của công ty, nhóm), thì sẽ dẫn đến suy giảm chất lượng phần mềm (do phụ thuộc quá nhiều vào con người)
- (4) Nếu không có chuẩn về làm tư liệu quy trình sản xuất phần mềm, thì những đặc tả không rõ ràng sẽ làm giảm chất lượng phần mềm
- (5) Nếu không kiểm thử tính đúng đắn của phần mềm ở từng giai đoạn mà chỉ kiểm ở giai đoạn cuối và phát hiện ra lỗi, thì thường bàn giao sản phẩm không đúng hạn
- (6) Nếu coi trọng việc lập trình hơn khâu thiết kế thì thường dẫn đến làm giảm chất lượng phần mềm
- (7) Nếu coi thường việc tái sử dụng phần mềm (software reuse), thì năng suất lao động sẽ giảm
- (8) Phần lớn trong quy trình phát triển phần mềm có nhiều thao tác do con người thực hiện, do vậy năng suất lao động thường bị giảm
- (9) Không chứng minh được tính đúng đắn của phần mềm, do vậy độ tin cậy của phần mềm sẽ giảm
- (10) Chuẩn về một phần mềm tốt không thể đo được một cách định lượng, do vậy không thể đánh giá được một hệ thống đúng đắn hay không
- (11) Khi đầu tư nhân lực lớn vào bảo trì sẽ làm giảm hiệu suất lao động của nhân viên
- (12) Công việc bảo trì kéo dài làm giảm chất lượng của tư liệu và ảnh hưởng xấu đến những việc khác
- (13) Quản lý dự án lỏng lẻo kéo theo quản lý lịch trình cũng không rõ ràng
- (14) Không có tiêu chuẩn để ước lượng nhân lực và dự toán sẽ làm kéo dài thời hạn và vượt kinh phí của dự án

Câu 6. Phân tích vòng đời phát triển phần mềm

Là khoảng thời gian tính từ khi phần mềm được đề xuất cho đến khi bỏ đi: cụ thể là từ khi được đặt hàng, phát triển, sử dụng đến khi bị loại bỏ.

❖ các giai đoạn: phân tích và đặc tả yêu cầu, thiết kế phần mềm, lập trình.

kiểm thử, cài đặt, bảo trì

❖ đặc điểm và các hoạt động chính của mỗi giai đoạn:

- phân tích và đặc tả yêu cầu: là bản đặc tả các dịch vụ mà hệ thống cung cấp, các ràng buộc để xây dựng vận hành hệ thống, là khâu đầu tiên trong quá trình xây dựng phần mềm

- Hoạt động chính của giai đoạn này là:

- + Phát hiện yêu cầu
- + Phân tích yêu cầu và thương lượng với khách hàng
- + Mô tả các yêu cầu
- + Mô tả hệ thống
- + Kiểm tra tính hợp lý của yêu cầu
- + Quản trị các yêu cầu
 - Thiết kế phần mềm: là thiết kế cấu hình phần cứng và cấu trúc phần mềm để có dc hệ thống thỏa mãn yêu cầu đặt ra

- Đặc điểm: chọn chiến lược cài đặt và quản lý dữ liệu

- + Tìm ra nguồn tài nguyên chung và cơ chế điều khiển truy nhập chung
- + Thiết kế cơ chế điều khiển thích hợp cho hệ thống, kể cả quản lý nhiệm vụ
- + Xem xét các điều kiện ràng buộc để xử lý ngoại

- Hoạt động chính của giai đoạn này là:

- + Thiết kế giao diện
- + Thiết kế chương trình
- + Thiết kế tập tin dữ liệu
 - Lập trình: là quá trình chuyển đổi từ thiết kế chi tiết sang mã lệnh. Lựa chọn ngôn ngữ lập trình phụ thuộc vào cấu hình máy, số lượng ngôn ngữ lập trình có sẵn, thói quen sử dụng ngôn ngữ lập trình. Người lập trình cần xác định thông tin tối thiểu cho 1 module chương trình
 - Kiểm thử: là 1 trong những giai đoạn quan trọng nhất phát triển phần mềm, là mấu chốt của đảm bảo chất lượng phần mềm. là tiến trình xem xét lại đặc tả, thiết kế, mã hóa nhằm phát hiện lỗi phần mềm. kiểm thử thành công khi phát hiện ra lỗi, kiểm thử ko phát hiện ra lỗi là kiểm thử dở

Một phép kiểm thử bao gồm: tên module kiểm thử, dữ liệu vào, dữ liệu r among muốn, dữ liệu ra thực tế, đánh giá chất lượng phần mềm qua 2 kết quả mong muốn và thực tế

- Cài đặt: lập kế hoạch cài đặt: chuyển đổi phần cứng, CSDL, công nghệ quản lý, phong cách làm việc...biến đổi dữ liệu từ hệ thống cũ sang mới, biên soạn tài liệu hệ thống
- Bảo trì: là pha cuối cùng của vòng đời gồm 2 hoạt động là quản lý hoạt động bảo trì và chuẩn hóa hoạt động bảo trì

Câu 7. Mô tả lịch sử phát triển của công nghệ phần mềm. trình bày các tiêu chuẩn để có 1 phần mềm tốt

gồm 3 giai đoạn

❖ Giai đoạn thứ nhất: (~ 1960):

- phần cứng thay đổi liên tục, theo tính chuyên dụng
- xử lý theo lô
- môi trường lập trình có tính cá nhân
- sản xuất đơn chiếc
- > ngôn ngữ: mã máy, đặc thù cho từng máy
- > lập trình: là 1 nghệ thuật theo bản năng, chưa có phương pháp

❖ Giai đoạn 2: (~ giữa thập kỷ 70)

- Phần mềm đa nhiệm, đa người sử dụng
- Hệ thống thời gian thực
- Xuất hiện lưu trữ trực tuyến
- > phần mềm trở nên phức tạp dẫn đến nhiều dự án thất bại
- > chi phí tăng cao dẫn đến nhu cầu tăng cao

Ngôn ngữ: có cấu trúc: PL1, Algol60, Fortran, COBOL

Lập trình: có phương pháp lập trình

Phát triển hệ thống: chưa có phương pháp, kinh nghiệm là chính

❖ Giai đoạn 3: (~ đến nay)

- Hệ thống phân tán
- Mạng cục bộ, toàn cầu
- Bộ vi xử lý phát triển mạnh
- > phần mềm ngày càng phức tạp

->sử dụng nhiều máy tính cá nhân, phần mềm nhúng

->số người dùng tăng nhanh

Ngôn ngữ: ngôn ngữ bậc cao, hướng đối tượng

Phát triển hệ thống: có phương pháp, công cụ tự động

Tiêu chuẩn đánh giá 1 phần mềm tốt: là phần mềm có chất lượng thỏa mãn:

Những yếu tố bên ngoài của 1 sản phẩm như: tốc độ nhanh, chạy ổn định, dễ sử dụng, dễ thích nghi với những thay đổi và một số yếu tố như: tính đúng đắn, tính bền vững, tính có thể mở rộng đc, tính sử dụng lại, tính tương thích , tính hiệu quả, tính bảo trì đc, tính tin cậy...

Những yếu tố chất lượng bên trong như tính dễ đọc, dễ hiểu, sản phẩm trực quan sinh động...

Câu 8. trình bày tóm tắt các loại hồ sơ (tài liệu) đi kèm sản phẩm phần mềm khi bàn giao với người sử dụng

1. Đặc tả hệ thống.
2. Kế hoạch dự án phần mềm.
 - a. Đặc tả yêu cầu phần mềm.
 - b. Bản mẫu thực hiện được hay "trên giấy".
3. Tài liệu người dùng sơ bộ
4. Đặc tả thiết kế.
 - a. Mô tả thiết kế dữ liệu.
 - b. Mô tả thiết kế kiến trúc.
 - c. Mô tả thiết kế module.
 - d. Mô tả thiết kế giao diện.
 - e. Mô tả sự vật (nếu kỹ thuật hướng sự vật được dùng).
5. Bản in chương trình gốc.
 - a. Chương trình nguồn.
 - b. Bản in chương trình nguồn (listing).
 - c. Bản mô tả thuật toán tương ứng với chương trình nguồn.
 - d. Kế hoạch và thủ tục kiểm thử.
 - e. Các trường hợp kiểm thử và kết quả ghi lại.
6. Tài liệu vận hành và cài đặt.

- a. Bản liệt kê các lỗi và cách xử lý.
- b. Bản liệt kê các thông số đặc trưng của hệ thống.
- 7. Chương trình thực hiện được.
 - a. Các module mã - thực hiện được.
 - b. Các module móc nối.
 - c. Chương trình đích lưu trữ trên vật mang tin.
- 8. Mô tả cơ sở dữ liệu.
 - a. Sơ đồ và cấu trúc tệp.
 - b. Nội dung ban đầu.
- 9. Tài liệu người sử dụng đã xây dựng.
 - a. Bản hướng dẫn sử dụng chi tiết.
 - b. Bản tóm tắt hướng dẫn sử dụng.
 - c. Các chương trình trợ giúp có liên quan.
- 10. Tài liệu bảo trì.
 - a. Báo cáo vấn đề phần mềm.
 - b. Yêu cầu bảo trì.
 - c. Trình tự thay đổi kỹ nghệ.
- 11. Các chuẩn và thủ tục cho kỹ thuật phần mềm .
- 12. Các tư liệu khác: hợp đồng, phiên bản, tài liệu pháp lý,...

Câu 9. trình bày đặc trưng cơ bản của mô hình gia tăng,so sánh ưu nhược điểm của mô hình gia tăng và mô hình bản mẫu nhanh

Đặc trưng cơ bản :

- _kết hợp mô hình tuần tự và ý tưởng lặp lại của chế bản mẫu
- _sản phẩm lỗi với những yêu cầu cơ bản nhất của các hệ thống đc phát triển
- _các chức năng với những yêu cầu khác đc phát triển thêm sau
- _lặp lại quy trình để hoàn thiện dần
- _mô hình đc đề xuất trên ý tưởng thay vì xây dựng và chuyển giao hệ thống 1 lần thì sẽ đc chia thành nhiều giai đoạn, tăng dần. mỗi giai đoạn là 1 phần kết quả của 1 chức năng đc yêu cầu
- _các yêu cầu của người sử dụng đc đánh thứ tự ưu tiên. Yêu cầu nào có thứ tự ưu tiên càng cao thì càng ở trong những giai đoạn phát triển sớm hơn

Mô hình gia tăng :

Ưu điểm : sau mỗi vòng tăng thì có thể chuyển giao kết quả thực hiện đc cho khách hàng nên các chức năng của hệ thống có gtheer nhìn thấy sớm hơn

Các vòng trc đóng vai trò là mẫu thử để giúp tìm hiểu thêm các yêu cầu ở những vòng tiếp theo

Những chức năng của hệ thống có thứ tự ưu tiên càng cao thì sẽ đc kiểm thử càng kỹ

Nhược điểm: – Cần phải có những khả năng thiết kế tốt và phương pháp tốt, để có thể hiểu rõ được yêu cầu và biết cách phân chi nó ra như thế nào cho hợp lý.

– Chi phí để phát triển theo phương pháp này là rất cao

Mô hình làm bản mẫu nhanh

Ưu điểm: Người sử dụng sớm hình dung ra chức năng và đặc điểm của hệ thống.

Phù hợp với: hệ thống rủi ro cao,yêu cầu không chắc chắn,giao diện chưa rõ rang,chiến lược cài đặt chưa rõ ràng

Nhược điểm:_khách hàng có thể cho rằng nguyên mẫu là hệ thống thực (mong đợi không thực tế về tiến triển dự án)

_người phát triển có sự chọn lựa ko tốt(phù hợp cho nguyên mẫu,nhưng ko phù hợp cho hệ thống thực)

_nguyên mẫu ko giống hoàn toàn hệ thống cuối cùng

Câu 10. trình bày đặc trưng cơ bản của mô hình hợp nhất.mô hình này thường ứng dụng trong dự án phần mềm nào

Đặc trưng cơ bản :_Mô hình hợp nhất sử dụng các kĩ thuật thể hệ thức 4 hay còn gọi là 4GT

_4GT bao gồm một phạm vi rộng các công cụ phần mềm có các điểm chung:

1,Cho phép người phát triển xác định một số đặc trưng của phần mềm ở mức cao

2,Tự động sinh ra mã chương trình gốc theo nhu cầu nhà phát triển

_tiến trình hợp nhất có thể nhìn dưới 2 góc nhìn khác nhau :

Quản lý(kinh tế chiến thuật, con người)

kỹ thuật(công nghệ,phương pháp,kiểm tra chất lượng)

Ưu điểm :giảm thời gian phát triển và tăng năng suất lao động

Nhược điểm :4GT khó dùng hơn ngôn ngữ lập trình,mã khó tối ưu khó bảo trì cho hệ thống lớn

Lĩnh vực ứng dụng hiện tại cho 4GT mới chỉ giới hạn vào các ứng dụng hệ thống tin nghiệp vụ, đặc biệt , việc phân tích thông tin và làm báo cáo là nhân tố chủ chốt cho các cơ sở dữ liệu lớn

Câu 11: Dựa trên phân tích cụ thể, hãy chỉ ra những ưu điểm của bản mẫu nhanh trong tiến trình phát triển phần mềm.

Mô hình bản mẫu

Qui trình được bắt đầu bằng việc thu thập yêu cầu với sự có mặt của đại diện của cả phía phát triển lẫn khách hàng nhằm định ra mục tiêu tổng thể của hệ thống phần mềm sau này, đồng thời ghi nhận tất cả những yêu cầu có thể biết được và sơ lược những nhóm yêu cầu nào cần phải được làm rõ.

Lợi ích của bản mẫu:

- Học bằng cách làm việc
- Tăng cường giao tiếp
- Tăng sự tham gia của người dùng vào dự án
- Làm rõ các yêu cầu chỉ biết một phần

Ưu điểm : phù hợp với

- Hệ thống rủi ro cao
- Yêu cầu không chắc chắn
- Giao diện chưa rõ ràng
- Chiến lược cài đặt chưa rõ ràng

Câu 12: Trình bày các đặc trưng cơ bản của mô hình gia tăng và mô hình xoắn ốc. So sánh ưu nhược điểm của mô hình gia tăng và mô hình xoắn ốc.

-Mô Hình Xoắn Ốc:

Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước.

Mô hình xoắn ốc được ưa chuộng cho các dự án lớn, đắt tiền và phức tạp.

Mô hình này sử dụng nhiều những giai đoạn tương tự như mô hình thác nước, về thứ tự, plan, đánh giá rủi ro, ...

Các pha trong quy trình phát triển xoắn ốc bao gồm:

Thiết lập mục tiêu: xác định mục tiêu cho từng pha của dự án.

Đánh giá và giảm thiểu rủi ro: rủi ro được đánh giá và thực hiện các hành động để giảm thiểu rủi ro.

Phát triển và đánh giá: sau khi đánh giá rủi ro, một mô hình xây dựng hệ thống sẽ được lựa chọn từ những mô hình chung.

Lập kế hoạch: đánh giá dự án và pha tiếp theo của mô hình xoắn ốc sẽ được lập kế hoạch.

Ưu điểm:

- Hạn chế rủi ro sớm
- Nhận được phản hồi (feedbacks) từ khách hàng sớm
- Dễ kiểm soát các mạo hiểm ở từng mức tiến hóa

Hạn chế:

- Khó thuyết phục khách hàng là phương pháp tiến hóa xoắn ốc có thể kiểm soát được
- Chưa được dùng rộng rãi như các mô hình tuyến tính hoặc chế thử.
- Chi phí cao và mất nhiều thời gian để hoàn thành dự án

Mô Hình Gia Tăng

- Những chức năng của hệ thống có thứ tự ưu tiên càng cao thì sẽ được kiểm thử càng kỹ.

Mô hình này được đề xuất dựa trên ý tưởng thay vì phải xây dựng và chuyển giao hệ thống một lần thì sẽ được chia thành nhiều giai đoạn, tăng dần. Mỗi giai đoạn là một phần kết quả của một chức năng được yêu cầu.

- Các yêu cầu của người sử dụng được đánh thứ tự ưu tiên. Yêu cầu nào có thứ tự ưu tiên càng cao thì càng ở trong những giai đoạn phát triển sớm hơn

Ưu điểm

- Sau mỗi lần tăng vòng thì có thể chuyển giao kết quả thực hiện được cho khách hàng nên các chức năng của hệ thống có thể nhìn thấy sớm hơn.
- Các vòng trước đóng vai trò là mẫu thử để giúp tìm hiểu thêm các yêu cầu ở những vòng tiếp theo.

Nhược điểm

- Cần lập plan và thiết kế tốt
- Cần một định nghĩa rõ ràng và đầy đủ của toàn bộ hệ thống trước khi nó có thể được chia nhỏ và được xây dựng từng bước

Câu 13: Mô hình xoắn ốc được cho là phù hợp trong phát triển các dự án phần mềm lớn. Hãy giải thích dựa trên đặc điểm của mô hình này.

-Mô Hình Xoắn Ốc:

Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước.

Mô hình xoắn ốc được ưa chuộng cho các dự án lớn, đắt tiền và phức tạp.

Mô hình này sử dụng nhiều những giai đoạn tương tự như mô hình thác nước, về thứ tự, plan, đánh giá rủi ro, ...

Các pha trong quy trình phát triển xoắn ốc bao gồm:

Thiết lập mục tiêu: xác định mục tiêu cho từng pha của dự án.

Đánh giá và giảm thiểu rủi ro: rủi ro được đánh giá và thực hiện các hành động để giảm thiểu rủi ro.

Phát triển và đánh giá: sau khi đánh giá rủi ro, một mô hình xây dựng hệ thống sẽ được lựa chọn từ những mô hình chung.

Lập kế hoạch: đánh giá dự án và pha tiếp theo của mô hình xoắn ốc sẽ được lập kế hoạch.

Ưu điểm:

- Hạn chế rủi ro sớm
- Nhận được phản hồi (feedbacks) từ khách hàng sớm
- Dễ kiểm soát các mạo hiểm ở từng mức tiến hóa

Hạn chế:

- Khó thuyết phục khách hàng là phương pháp tiến hóa xoắn ốc có thể kiểm soát được
- Chưa được dùng rộng rãi như các mô hình tuyến tính hoặc chế thử.
- Chi phí cao và mất nhiều thời gian để hoàn thành dự án

Câu 14: Trình bày mô hình thác nước và mô hình xoắn ốc. Nêu ưu và nhược điểm và sự khác nhau của hai mô hình đó?

-Mô Hình Xoắn Ốc:

Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước.

Mô hình xoắn ốc được ưa chuộng cho các dự án lớn, đắt tiền và phức tạp.

Mô hình này sử dụng nhiều những giai đoạn tương tự như mô hình thác nước, về thứ tự, plan, đánh giá rủi ro, ...

Các pha trong quy trình phát triển xoắn ốc bao gồm:

Thiết lập mục tiêu: xác định mục tiêu cho từng pha của dự án.

Đánh giá và giảm thiểu rủi ro: rủi ro được đánh giá và thực hiện các hành động để giảm thiểu rủi ro.

Phát triển và đánh giá: sau khi đánh giá rủi ro, một mô hình xây dựng hệ thống sẽ được lựa chọn từ những mô hình chung.

Lập kế hoạch: đánh giá dự án và pha tiếp theo của mô hình xoắn ốc sẽ được lập kế hoạch.

Ưu điểm:

- Hạn chế rủi ro sớm
- Nhận được phản hồi (feedbacks) từ khách hàng sớm
- Dễ kiểm soát các mạo hiểm ở từng mức tiến hóa

Hạn chế:

- Khó thuyết phục khách hàng là phương pháp tiến hóa xoắn ốc có thể kiểm soát được
- Chưa được dùng rộng rãi như các mô hình tuyến tính hoặc chế thử.
- Chi phí cao và mất nhiều thời gian để hoàn thành dự án

-Mô Hình Thác Nước

Là mô hình cổ điển, Phương pháp áp dụng 1 lần, Điều khiển hiệu quả, Phạm vi giới hạn của vòng lặp, vòng đời dài, không thích hợp với hệ thống không rõ ràng.

Đây được coi như là mô hình phát triển phần mềm đầu tiên được sử dụng Mô hình này áp dụng tuần tự các giai đoạn của phát triển phần mềm. Đầu ra của giai đoạn trước là đầu vào của giai đoạn sau. Giai đoạn sau chỉ được thực hiện khi giai đoạn trước đã kết thúc. Đặc biệt không được quay lại giai đoạn trước để xử lý các yêu cầu khi muốn thay đổi.

Ưu điểm:

- Chỉ phù hợp với các dự án nhỏ
- Dễ sử dụng, dễ tiếp cận, dễ quản lý
- Sản phẩm phát triển theo các giai đoạn được xác định rõ ràng

Nhược điểm:

- Ít linh hoạt, phạm vi điều chỉnh hạn chế
- Không phù hợp với dự án lớn
- Thời gian thực hiện lâu

Câu 15: Trình bày các đặc trưng cơ bản của mô hình làm bản mẫu nhanh. So sánh ưu nhược điểm của mô hình làm bản mẫu nhanh và mô hình thác nước.

Mô hình bản mẫu nhanh:

Tập hợp yêu cầu, Thiết kế nhanh, Xây dựng bản mẫu, Đánh giá của khách hàng, Làm mịn, Quay lại thiết kế nhanh để điều chỉnh, Xây dựng sản phẩm,..

Quy trình được bắt đầu bằng việc thu thập yêu cầu với sự có mặt của đại diện của cả phía phát triển lẫn khách hàng nhằm định ra mục tiêu tổng thể của hệ thống phần mềm sau này, đồng thời ghi nhận tất cả những yêu cầu có thể biết được và sơ lược những nhóm yêu cầu nào cần phải được làm rõ.

Lợi ích của bản mẫu:

- Học bằng cách làm việc
- Tăng cường giao tiếp
- Tăng sự tham gia của người dùng vào dự án
- Làm rõ các yêu cầu chỉ biết một phần

Ưu điểm : phù hợp với

- Hệ thống rủi ro cao
- Yêu cầu không chắc chắn
- Giao diện chưa rõ ràng
- Chiến lược cài đặt chưa rõ ràng

MH thác nước

Ưu điểm:

- Chỉ phù hợp với các dự án nhỏ hoặc Yêu cầu xác định
- Dễ sử dụng, dễ tiếp cận, dễ quản lý

Nhược điểm:

- Ít linh hoạt, phạm vi điều chỉnh hạn chế
- Không phù hợp với dự án lớn
- Thời gian thực hiện lâu

Câu 16: Trình bày các đặc trưng cơ bản của mô hình thác nước và mô hình chữ

V. So sánh ưu nhược điểm của mô hình mô hình thác nước và mô hình chữ V.

-Mô Hình Thác Nước

Là mô hình cổ điển, Phương pháp áp dụng 1 lần, Điều khiển hiệu quả, Phạm vi giới hạn của vòng lặp, vòng đời dài, không thích hợp với hệ thống không rõ ràng.

Đây được coi như là mô hình phát triển phần mềm đầu tiên được sử dụng Mô hình này áp dụng tuần tự các giai đoạn của phát triển phần mềm. Đầu ra của giai đoạn trước là đầu vào của giai đoạn sau. Giai đoạn sau chỉ được thực hiện khi giai đoạn trước đã kết thúc. Đặc biệt không được quay lại giai đoạn trước để xử lý các yêu cầu khi muốn thay đổi.

Ưu điểm:

- Chỉ phù hợp với các dự án nhỏ
- Dễ sử dụng, dễ tiếp cận, dễ quản lý
- Sản phẩm phát triển theo các giai đoạn được xác định rõ ràng

Nhược điểm:

- Ít linh hoạt, phạm vi điều chỉnh hạn chế
- Không phù hợp với dự án lớn
- Thời gian thực hiện lâu
- MH chữ V
- Các tiến trình kiểm thử được thêm vào
- Kết nối kiểm thử với phân tích và thiết kế
- Thích hợp với những trường hợp bài toán không nhất quán
- Yêu cầu được xác định rõ ràng, được ghi chép và cố định rõ ràng.
- Xác định sản phẩm ổn định.
- Công nghệ không thay đổi và được hiểu rõ bởi nhóm dự án.
- Không có yêu cầu không rõ ràng hoặc không xác định.
- Dự án ngắn.

Ưu điểm

Đây là một mô hình có tính kỷ luật cao và các giai đoạn được hoàn thành cùng một lúc.

Hoạt động tốt cho các dự án nhỏ, khi các yêu cầu được hiểu rất rõ.

Đơn giản và dễ hiểu và dễ sử dụng.

Dễ quản lý. Mỗi giai đoạn có phân phối cụ thể và quy trình đánh giá.

Nhược điểm Khó quản lý kiểm soát rủi ro, rủi ro cao

Không phải là một mô hình tốt cho các dự án phức tạp và hướng đối tượng.

Mô hình kém cho các dự án dài và đang diễn ra.

Không thích hợp cho các dự án có nguy cơ thay đổi yêu cầu trung bình đến cao.

Khi ứng dụng đang ở giai đoạn thử nghiệm, rất khó để quay lại và thay đổi chức năng.

Câu 17: Trình bày các đặc trưng cơ bản của mô hình xoắn ốc. So sánh ưu nhược điểm của mô hình xoắn ốc và mô hình làm bản mẫu nhanh.?

-Mô Hình Xoắn Ốc:

Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước.

Mô hình xoắn ốc được ưa chuộng cho các dự án lớn, đắt tiền và phức tạp.

Mô hình này sử dụng nhiều những giai đoạn tương tự như mô hình thác nước, về thứ tự, plan, đánh giá rủi ro, ...

Các pha trong quy trình phát triển xoắn ốc bao gồm:

Thiết lập mục tiêu: xác định mục tiêu cho từng pha của dự án.

Đánh giá và giảm thiểu rủi ro: rủi ro được đánh giá và thực hiện các hành động để giảm thiểu rủi ro.

Phát triển và đánh giá: sau khi đánh giá rủi ro, một mô hình xây dựng hệ thống sẽ được lựa chọn từ những mô hình chung.

Lập kế hoạch: đánh giá dự án và pha tiếp theo của mô hình xoắn ốc sẽ được lập kế hoạch.

Ưu điểm:

- Hạn chế rủi ro sớm
- Nhận được phản hồi (feedbacks) từ khách hàng sớm
- Dễ kiểm soát các mạo hiểm ở từng mức tiến hóa

Hạn chế:

- Khó thuyết phục khách hàng là phương pháp tiến hóa xoắn ốc có thể kiểm soát được
- Chưa được dùng rộng rãi như các mô hình tuyến tính hoặc chế thử.

– Chi phí cao và mất nhiều thời gian để hoàn thành dự án

-Mô Hình Bản Mẫu Nhanh :

Ưu điểm:

- Học bằng cách làm việc
- Tăng cường giao tiếp
- Tăng sự tham gia của người dùng vào dự án
- Làm rõ các yêu cầu chỉ biết một phần

Nhược điểm :

- Hệ thống rủi ro cao
- Yêu cầu không chắc chắn
- Giao diện chưa rõ ràng
- Chiến lược cài đặt chưa rõ ràng

Câu 18 : Mô hình hợp nhất được cho là phù hợp trong phát triển các dự án phần mềm lớn. Hãy giải thích dựa trên đặc điểm của mô hình này.

- Mô hình hợp nhất :

Mô hình hợp nhất sử dụng các kỹ thuật thế hệ 4 (4GT) : tập hợp các công cụ cho phép kết hợp đặc tính phần mềm ở mức cao, sau đó tự động sinh mã nguồn dựa theo đặc tả đó.

Các công cụ 4GT điển hình : ngôn ngữ phi thủ tục cho truy vấn CSDL, tạo báo cáo, xử lý dữ liệu, tương tác màn hình, tạo mã nguồn, khả năng đồ họa bậc cao, khả năng bảng tính, khả năng duyệt web.

Từ thu thập yêu cầu cho đến sản phẩm : đối thoại giữa khách hàng và người phát triển là rất quan trọng

Không nên bỏ qua khâu thiết kế : 4GT chỉ áp dụng để triển khai thiết kế qua ngôn ngữ lập trình thế hệ 4 (4GL)

Tiến trình hợp nhất có thể được nhìn dưới 2 góc nhìn khác nhau :

- + Góc nhìn quản lý : quan tâm đến lĩnh vực kinh tế, con người, chiến thuật
- + Góc nhìn kỹ thuật : quan tâm đến công nghệ, kiểm tra chất lượng, phương pháp.

Ưu điểm: Giảm thời gian phát triển và tăng năng suất lao động

Nhược điểm : 4Gt khó dung hơn ngôn ngữ lập trình, mã khó tối ưu và khó bảo trì hệ thống lớn nên cần kỹ năng của kỹ sư phần mềm.

Câu 19 : Dựa trên phân tích cụ thể, hãy chỉ ra những ưu điểm của mô hình bản mẫu nhanh trong tiến trình phát triển phần mềm.

Giống câu 11

Câu 20 : . Trình bày các đặc trưng cơ bản của mô hình làm bản mẫu nhanh. So sánh ưu nhược điểm của mô hình làm bản mẫu nhanh và mô hình thác nước.

Giống câu 15

Câu 21: Dựa trên phân tích cụ thể, hãy chỉ ra những ưu điểm của mô hình xoắn ốc trong tiến trình phát triển phần mềm.

Mô hình xoắn ốc (tiếng Anh: *spiral model*) là quy trình phát triển định hướng rủi ro cho các dự án phần mềm. Kết hợp của thể mạnh của các mô hình khác và giải quyết khó khăn của các mô hình trước còn tồn tại. Dựa trên các mô hình rủi ro riêng biệt của mỗi dự án, mô hình xoắn ốc đưa ra cách áp dụng các yếu tố của một hoặc nhiều mô hình xử lý, chẳng hạn như mô hình gia tốc, mô hình thác nước hoặc mô hình tạo mẫu tiến hóa.

Ưu điểm[sửa | sửa mã nguồn]

Là mô hình hội tụ các tính năng tốt và khắc phục các yếu điểm của nhiều mô hình phát triển khác gặp phải.

Giám sát dự án dễ dàng và hiệu quả

Rất phù hợp với dự án có nguy cơ cao và giảm thiểu rủi ro, đối phó với những thay đổi trong quá trình thực hiện dự án

Dự đoán về thời hạn và chi phí sát với thực tế

Nhược điểm[sửa | sửa mã nguồn]

- Phân tích rủi ro khá tốn kém, chủ yếu áp dụng cho dự án lớn, có tiềm lực về tài chính
- Yêu cầu thay đổi thời xuyên dẫn đến lặp vô hạn, phức tạp, cần có đội ngũ chuyên gia về phân tích rủi ro
- Chưa được áp dụng rộng rãi như mô hình thác nước, nguyên mẫu.

Câu22: Phân tích các hoạt động chính trong giai đoạn phân tích phần mềm. Tại sao việc xác định nhu cầu người sử dụng và phân tích tính khả thi là những bước đầu tiên trong quá trình phân tích phần mềm?

các hoạt động trong phân tích phần mềm

Đặc tả phần mềm

- Công việc đầu tiên cần phải làm trong quá trình xây dựng phần mềm là gì?
- Tầm quan trọng của việc đặc tả phần mềm.

Đặc tả phần mềm (hay còn gọi là kỹ thuật xác định yêu cầu) là quy trình tìm hiểu và định nghĩa những dịch vụ nào được yêu cầu và các ràng buộc trong quá trình vận hành và xây dựng hệ thống.

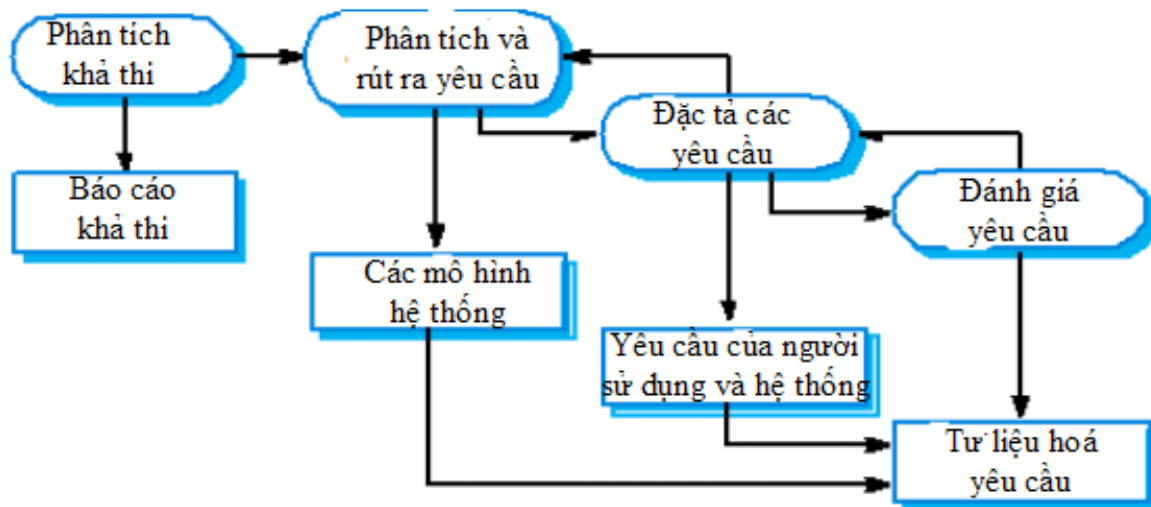
Quy trình xác định yêu cầu bao gồm bốn pha chính:

- Nghiên cứu khả thi: Nghiên cứu khả thi giúp xác định những yêu cầu của người sử dụng có thoả mãn những công nghệ hiện tại hay không. Về góc độ kinh doanh, nghiên cứu khả thi nhằm xác định hệ thống đưa ra có mang lại lợi nhuận không. Việc nghiên cứu khả thi nên được thực hiện một cách nhanh chóng và không quá tốn kém. Kết quả của việc nghiên cứu khả thi sẽ xác định có nên tiếp tục xây dựng hệ thống nữa hay không.
- Phân tích và rút ra các yêu cầu: đây là quy trình đưa ra các yêu cầu hệ thống thông qua một số phương pháp như: quan sát hệ thống hiện tại, phỏng vấn và thảo luận với người sử dụng, phân tích nhiệm vụ, phân tích tài liệu hoặc hệ thống cũ ... Trong pha này, chúng ta có thể phải xây dựng một hoặc nhiều mô hình hệ thống và các mẫu thử.
- Đặc tả yêu cầu: Pha này sẽ tư liệu hoá những thông tin thu thập được. Có hai loại yêu cầu cần được xác định:

* Yêu cầu của người sử dụng: là những yêu cầu bằng ngôn ngữ tự nhiên bổ sung thêm cho các biểu đồ của các dịch vụ mà hệ thống cung cấp và các ràng buộc vận hành của nó. Kiểu yêu cầu này được viết bởi người sử dụng.

* Yêu cầu hệ thống: là những tài liệu có cấu trúc mô tả chi tiết về các chức năng, dịch vụ và các ràng buộc vận hành của hệ thống. Yêu cầu hệ thống sẽ định nghĩa những gì cần phải xây dựng, cho nên nó có thể trở thành bản hợp đồng giữa khách hàng và nhà thầu.

- Đánh giá yêu cầu: pha này sẽ kiểm tra lại các yêu cầu xem chúng có đúng thực tế hay không, có thống nhất không, có đầy đủ không. Nếu phát hiện ra lỗi thì ta phải chỉnh sửa các lỗi này.



Hình 2.6: Quy trình xác định yêu cầu

Thiết kế phần mềm và cài đặt

- Bỏ qua giai đoạn thiết kế, sau khi đặc tả và phân tích yêu cầu, có thể thực hiện cài đặt hệ thống ngay được không?

- Vai trò của bản thiết kế đối với giai đoạn cài đặt là gì?

Thiết kế phần mềm là quá trình thiết kế cấu trúc phần mềm dựa trên những tài liệu đặc tả. Hoạt động thiết kế bao gồm những công việc chính sau:

- Thiết kế kiến trúc: Các hệ thống con cấu thành lên hệ thống cần xây dựng và mối quan hệ giữa chúng được xác định và tư liệu hoá.

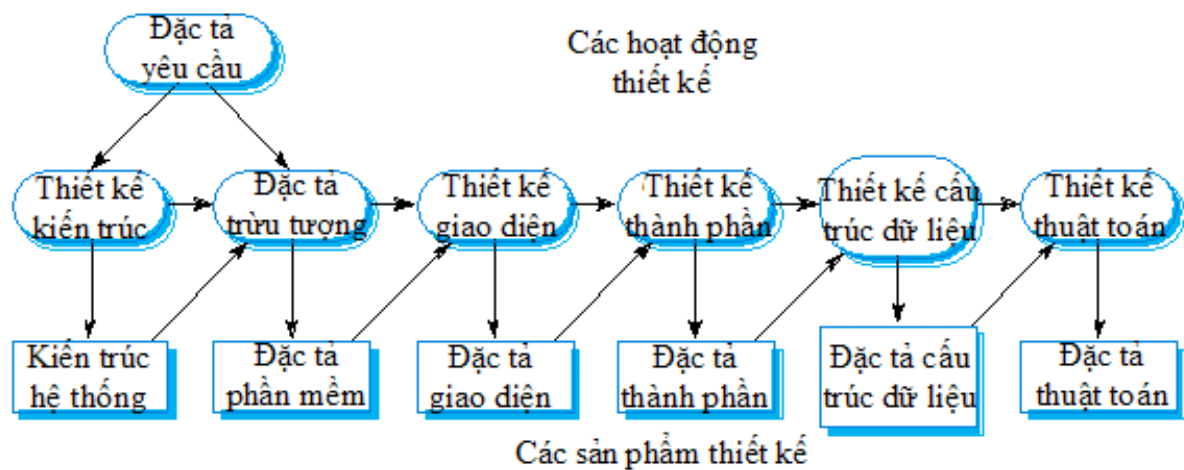
- Đặc tả trừu tượng: với mỗi hệ thống con, phải có một bản đặc tả về các dịch vụ của nó và những ràng buộc khi nó vận hành.

- Thiết kế giao diện: với mỗi hệ thống con, các giao diện của nó với những hệ thống con khác phải được thiết kế và tư liệu hoá.

- Thiết kế thành phần: các dịch vụ cung cấp cho các thành phần khác và các giao diện tương tác với chúng phải được thiết kế.

- Thiết kế cấu trúc dữ liệu: cấu trúc dữ liệu được sử dụng để cài đặt hệ thống phải được thiết kế một cách chi tiết và cụ thể.

- Thiết kế thuật toán: Các thuật toán được sử dụng để cung cấp các dịch vụ phải được thiết kế chi tiết và chính xác.



Hình 2.7: Mô hình chung của quy trình thiết kế

Cài đặt là quy trình chuyển đổi từ tài liệu đặc tả hệ thống thành một hệ thống thực, có thể vận hành được và phải loại bỏ các lỗi của chương trình.

Lập trình là một hành động cá nhân, không có quy trình lập trình chung. Người lập trình phải thực hiện một số kiểm thử để phát hiện ra lỗi trong chương trình và loại bỏ nó trong quy trình gỡ lỗi.

Đánh giá phần mềm

- Sau khi cài đặt phần mềm, chúng ta có thể chuyển giao ngay cho người sử dụng được không?

- Vai trò của việc đánh giá phần mềm là gì?

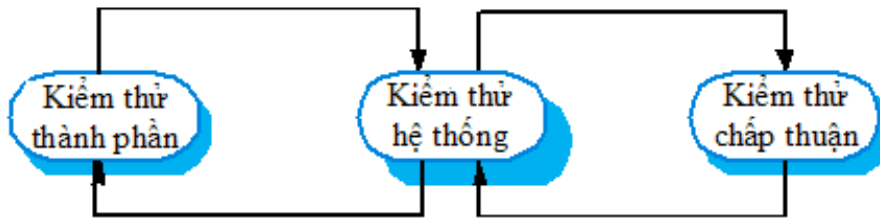
Đánh giá phần mềm hay còn gọi là thẩm tra và đánh giá (V&V - Verification and validation) được sử dụng để chỉ ra rằng hệ thống đã thực hiện theo đúng các đặc tả và thoả mãn mọi yêu cầu của khách hàng.

Đánh giá phần mềm bao gồm các công đoạn: kiểm tra, xem xét lại, và kiểm thử hệ thống. Kiểm thử hệ thống tức là cho hệ thống thực hiện trên những trường hợp có dữ liệu thật được lấy từ tài liệu đặc tả hệ thống. Quy trình kiểm thử gồm các pha sau:

- Kiểm thử thành phần (đơn vị): các thành phần được kiểm thử một cách độc lập, thành phần có thể là một chức năng hoặc một đối tượng hoặc một nhóm các thực thể gắn kết với nhau.

- Kiểm thử hệ thống: kiểm thử toàn bộ hệ thống.

- Kiểm thử chấp thuận: kiểm thử trên dữ liệu của khách hàng để kiểm tra hệ thống có đáp ứng tất cả các yêu cầu của khách hàng hay không.



Hình 2.8: Quy trình kiểm thử

Khi chuyển giao hệ thống cho khách hàng thì quy trình kiểm thử beta sẽ được thực hiện. Khách hàng sẽ thông báo các lỗi cho đội dự án. Những lỗi này sẽ được chỉnh sửa và tiếp tục kiểm thử beta hoặc chuyển giao thực sự cho khách hàng.

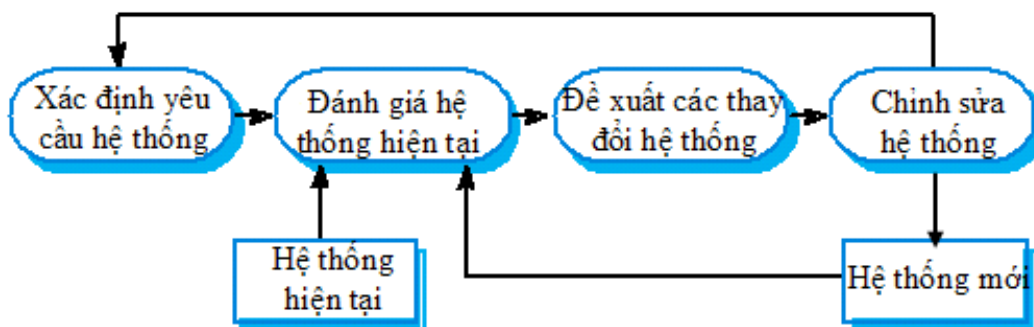
Cải tiến phần mềm



Đặt vấn đề

- Sau khi chuyển giao phần mềm cho khách hàng, thì mọi công việc đã kết thúc chưa?
- Cải tiến phần mềm để làm gì?
- Tại sao không xây dựng hệ thống mới mà lại cải tiến hệ thống cũ?

Khi các yêu cầu hệ thống thay đổi theo sự thay đổi của các yêu cầu nghiệp vụ thì phần mềm phải cải tiến và thay đổi để hỗ trợ khách hàng. Thông thường chi phí để bảo trì và cải tiến thường đắt hơn nhiều so với chi phí xây dựng phần mềm.



Hình 2.9: Cải tiến hệ thống

_ phải đánh giá vì

Câu 23: Trình bày nguyên tắc thiết kế giao diện người dùng? Theo bạn nguyên tắc nào quan trọng nhất? Vì sao?

Các nguyên tắc thiết kế giao diện

Thiết kế giao diện phải phụ thuộc vào yêu cầu, kinh nghiệm và khả năng của người sử dụng hệ thống.

Người thiết kế cũng nên quan tâm đến những giới hạn vật lý và tinh thần của con người và nên nhận ra rằng con người luôn có thể gây ra lỗi.

Không phải tất cả các nguyên tắc thiết kế giao diện đều có thể được áp dụng cho tất cả các giao diện. Sau đây là các nguyên tắc thiết kế giao diện:

- Sự quen thuộc của người sử dụng: giao diện phải được xây dựng dựa trên các thuật ngữ và các khái niệm mà người sử dụng có thể hiểu được hơn là những khái niệm liên quan đến máy tính. Ví dụ: hệ thống văn phòng nên sử dụng các khái niệm như thư, tài liệu, cặp giấy ... mà không nên sử dụng những khái niệm như thư mục, danh mục ...
- Thống nhất: hệ thống nên hiển thị ở mức thống nhất thích hợp. Ví dụ: các câu lệnh và menu nên có cùng định dạng ...
- Tối thiểu hoá sự bất ngờ: nếu một yêu cầu được xử lý theo cách đã biết trước thì người sử dụng có thể dự đoán các thao tác của những yêu cầu tương tự.
- Khả năng phục hồi: hệ thống nên cung cấp một số khả năng phục hồi từ lỗi của người sử dụng và cho phép người sử dụng khôi phục lại từ chỗ bị lỗi. Khả năng này bao gồm cho phép làm lại, hỏi lại những hành động như xoá, huỷ ...
- Hướng dẫn người sử dụng: như hệ thống trợ giúp, hướng dẫn trực tuyến ...
- Tính đa dạng: hỗ trợ nhiều loại tương tác cho nhiều loại người sử dụng khác nhau. Ví dụ: nên hiển thị phông chữ lớn với những người cận thị.

Tương tác giữa người sử dụng và hệ thống được chia thành 5 loại sau:

- Vận hành trực tiếp
- Lựa chọn menu
- Điền vào biểu mẫu (Form)
- Ngôn ngữ ra lệnh
- Ngôn ngữ tự nhiên

* nguyên tắc quan trọng nhất là 1 và 2 vì giao diện phải được xây dựng dựa trên các thuật ngữ và các khái niệm mà người sử dụng có thể hiểu được hơn là những khái niệm liên quan đến máy tính. Ví dụ: hệ thống văn phòng nên sử dụng các khái niệm như thư, tài liệu, cặp giấy ... mà không nên sử dụng những khái niệm như thư mục, danh mục ... và - Khả năng phục hồi: hệ thống nên cung cấp một số khả năng phục hồi từ lỗi của người sử dụng và cho phép người sử dụng khôi phục lại từ chỗ bị lỗi.

Câu 24: Khái niệm kiểm thử. So sánh ưu nhược điểm của phương pháp kiểm thử trên bàn (kiểm thử tĩnh) và kiểm thử trên máy (kiểm thử động).

Kiểm thử phần mềm (kiểm tra, thử nghiệm) là một cuộc kiểm tra được tiến hành để cung cấp cho các bên liên quan thông tin về chất lượng của sản phẩm hoặc dịch vụ được kiểm thử.^[1] Kiểm thử có thể cung cấp cho doanh nghiệp một quan điểm, một cách nhìn độc lập về phần mềm để từ đó cho phép đánh giá và thấu hiểu được những rủi ro trong quá trình triển khai phần mềm.

Trong kỹ thuật kiểm thử không chỉ giới hạn ở việc thực hiện một chương trình hoặc ứng dụng với mục đích đi tìm các lỗi phần mềm (bao gồm các lỗi và các thiếu sót) mà còn là một quá trình phê chuẩn và xác minh một chương trình máy tính / ứng dụng / sản phẩm.

- *ưu ,nhược điểm của **Kiểm thử tĩnh** (Static Testing)

Là phương pháp kiểm thử phần mềm đòi hỏi phải duyệt lại các yêu cầu và các đặc tả bằng tay, thông qua việc sử dụng giấy, bút để kiểm tra logic, lần từng chi tiết mà không cần chạy chương trình. Kiểu kiểm thử này thường được sử dụng bởi chuyên viên thiết kế người mà viết mã lệnh một mình.

Ưu điểm: Kiểm thử tĩnh cũng có thể được tự động hóa. Nó sẽ thực hiện kiểm tra toàn bộ bao gồm các chương trình được phân tích bởi một trình thông dịch hoặc biên dịch mà xác nhận tính hợp lệ về cú pháp của chương trình.

Nhược điểm: thủ công ,mất nhiều thời gian ,tốn nhiều nhân lực

- **Kiểm thử động** (Dynamic Testing)

Là phương pháp kiểm thử phần mềm thông qua việc dùng máy chạy chương trình để điều tra trạng thái tác động của chương trình. Đó là kiểm thử dựa trên các ca kiểm thử xác định bằng sự thực hiện của đối tượng kiểm thử hay chạy các chương trình. Kiểm thử động kiểm tra cách thức hoạt động của mã lệnh, tức là kiểm tra sự phản ứng vật lý từ hệ thống tới các biến luôn thay đổi theo thời gian.

Ưu điểm;-Trong kiểm thử động, phần mềm phải thực sự được biên dịch và chạy. Kiểm thử động thực sự bao gồm làm việc với phần mềm, nhập các giá trị đầu vào và kiểm tra xem liệu đầu ra có như mong muốn hay không. Các phương pháp kiểm thử động gồm có kiểm thử Unit – Unit Tests, Kiểm thử tích hợp – Intergration Tests, Kiểm thử hệ thống –System Tests, và Kiểm thử chấp nhận sản phẩm – Acceptance Tests.

-kiểm thử độ chính xác cao

-chu trình kiểm tra diễn ra trong thời gian ngắn

-cần ít nhân lực

* Nhược điểm: -chi phí cao cho việc chuyển giao công nghệ và đào tạo nhân viên

-tốn chi phí và thời gian cho việc tạo ra kịch bản kiểm thử và phát triển công cụ kiểm thử tự động.

Câu 25: Các lỗi có thể mắc phải trong quá trình cài đặt phần mềm là gì? Nguyên nhân? Theo bạn lỗi nào ảnh hưởng tới việc triển khai phần mềm nhất? Vì sao?

Câu 26: Chất lượng của thiết kế được thể hiện qua các yếu tố gì? Các yếu tố đó có vai trò như thế nào trong thiết kế phần mềm?

Chất lượng của thiết kế được thể hiện qua các yếu tố: Tính đúng đắn, tính tiến hóa, tính tiện dụng, tính hiệu quả, tính tương thích,

Vai trò của các yếu tố trong thiết kế phần mềm:

1. Tính đúng đắn: Phần mềm phải thực hiện đầy đủ và chính xác các yêu cầu của người dùng

2. Tính tiến hóa: Cho phép người dùng có thể khai báo các thay đổi về quy định của phần mềm theo các thay đổi trong thế giới thực

VD: - Cách tính trung bình kiểm tra, trung bình môn

- Cách tính lương dựa vào bảng chấm công

- Công thức tính tiền phạt

- Công thức tính tiền điện

3. Tính tiện dụng: Dễ học, dễ sử dụng, có giao diện trực quan, tự nhiên

4. Tính hiệu quả: Sử dụng tối ưu các tài nguyên (CPU, bộ nhớ, ...) --> sử dụng được trên cấu hình tối thiểu

5. Tính tương thích: Trao đổi dữ liệu được với các phần mềm khác có liên quan VD: - Nhập danh mục sách từ tập tin Excel - Xuất danh sách điểm thành tập tin Excel - Gửi báo cáo tổng kết thông qua Email bằng phần mềm WinFax

Câu 27: Trình bày các giai đoạn để thiết kế một sản phẩm phần mềm? Liệt kê các phương pháp thiết kế phần mềm hiện nay mà bạn biết?

Các giai đoạn để thiết kế một sản phẩm phần mềm:

1. Xác định yêu cầu:

Tiến hành ngay sau khi có nhu cầu về việc xây dựng phần mềm

Mục tiêu: Xác định chính xác yêu cầu đặt ra cho phần mềm sẽ xây dựng

Kết quả nhận: Thông tin và hoạt động của thế giới thực

Kết quả chuyển giao: Danh sách các yêu cầu (công việc sẽ thực hiện trên máy tính) cùng với các thông tin mô tả chi tiết về các yêu cầu (cách thực hiện công việc trong thế giới thực)

2. Phân tích: Tiến hành ngay sau khi kết thúc việc xác định yêu cầu Mục tiêu: Mô tả lại thế giới thực thông qua các mô hình (mô hình hóa thế giới thực) trước khi thiết kế --> thường là các sơ đồ

Kết quả nhận: Danh sách các yêu cầu cùng với thông tin liên quan

Kết quả chuyển giao:

- Mô hình dữ liệu: hệ thống các loại thông tin được sử dụng trong thế giới thực cùng với mối quan hệ giữa chúng
- Mô hình xử lý: hệ thống các công việc của thế giới thực cùng với quan hệ giữa chúng
- Các mô hình khác: không gian, thời gian, con người, ... nếu cần thiết (mô hình trạng thái, mô hình chức năng, sơ đồ tổ chức, ...) --> các mô hình thể hiện ở mức quan niệm (conceptual models)

3. Thiết kế: Tiến hành ngay sau khi kết thúc việc phân tích Mục tiêu: Mô tả các thành phần của phần mềm (mô hình phần mềm) trước khi cài đặt

Kết quả nhận: Mô hình thế giới thực (mô hình quan niệm)

Kết quả chuyển giao:

- Mô tả thành phần giao diện: các hàm nhập/ xuất, các hình thức nhập/ xuất, cấu trúc dữ liệu nhập/ xuất
- Mô tả thành phần xử lý: các hàm kiểm tra, xử lý
- Mô tả thành phần dữ liệu: các hàm đọc / ghi, tổ chức lưu trữ trên bộ nhớ phụ

4. Cài đặt:

Tiến hành ngay sau khi kết thúc việc thiết kế Mục tiêu: Tạo lập phần mềm mong muốn theo đúng yêu cầu

Kết quả nhận: Mô hình phần mềm

Kết quả chuyển giao: chương trình nguồn của phần mềm --> hệ thống các hàm cùng với cấu trúc dữ liệu nhập / xuất, CSDL tương ứng (nếu có) và chương trình thực hiện được trên máy tính (chương trình nguồn đã được biên dịch)

5. Kiểm thử: Tiến hành ngay sau khi có kết quả từng phần của việc lập trình Mục tiêu: Tăng độ tin cậy của phần mềm

Kết quả nhận: Danh sách các yêu cầu, Mô hình phần mềm ,Phần mềm

Kết quả chuyển giao: Phần mềm với độ tin cậy cao (đã sửa lỗi)

- **Các phương pháp thiết kế phần mềm hiện nay:**

- Phương pháp theo mô hình thác nước
- Phương pháp theo Mô hình làm bản mẫu
- Phương pháp theo Mô hình RAD (Rapid Application Development)
- Phương pháp theo Mô hình xoắn ốc

Câu 28: Nêu các nguyên tắc thiết kế giao diện người dùng ? Theo bạn nguyên tắc nào quan trọng nhất? Vì sao?

Các nguyên tắc thiết kế giao diện người dùng:

1: hệ thống dễ sử dụng dựa vào các tiêu chuẩn:

a. Tính thân thiện:

- Các chức năng dễ hiểu
- Hệ thống phát hiện ngay những sai sót và lỗi bất cần của người sử dụng.
- Dự trù sẵn những hành động khi có sự cố như lỗi kỹ thuật (cúp điện, hư đĩa cứng), lỗi nhập liệu sai, ...
- Uyển chuyển, linh hoạt đáp ứng được nhiều người dùng.
- Hoạt động theo trình tự tự nhiên của người dùng -> giống như quy trình nghiệp vụ của người dùng trong thế giới thực. Để đánh giá tính thân thiện của hệ thống phải trả lời các câu hỏi:

1. Tôi đang ở đâu trong hệ thống?
2. Tôi đã đến đây như thế nào?
3. Tôi có thể làm gì tại đây?
4. Sau đây tôi có thể đi đến đâu?

b. Tính Ergonomic: (nghiên cứu sự tương quan giữa người và môi trường làm việc).

- Màu sắc: hấp dẫn người sử dụng ảnh hưởng đối với thị lực của người dùng.

- Vị trí của các ô dữ liệu/nút chức năng ảnh hưởng đến năng suất.
 - Cách giao tiếp với hệ thống hỗ trợ cho tính chất thân thiện với người sử dụng.
- 2: tính nhất quán của hệ thống Thể hiện qua nhiều chuẩn:
- Chuẩn về dữ liệu: tên gọi, độ dài của dữ liệu khi thực hiện trên mô hình, hoặc thể hiện của dữ liệu, cách trình bày.
 - Chuẩn về mã hóa: liên quan đến thể hiện tiếng Việt, tiếng Anh phải nhất quán trên tất cả các màn hình.
 - Chuẩn về cấu trúc hệ thống: cách phân chia môđun trên hệ thống, hệ thống thực đơn,
 - ...
 - Chuẩn về sửa lỗi: liên quan đến sửa lỗi gắn trực tiếp vào hệ thống.

Từ các tiêu chí này thì một hệ thống thông tin có 3 loại giao diện:

1. Giao diện đầu vào: màn hình nhập dữ liệu (Thêm – Xóa – Sửa)
2. Giao diện đầu ra: các kết xuất ra màn hình hay máy in (các biểu mẫu báo cáo, thống kê).
3. Giao diện tương tác giữa người dùng với hệ thống: những thao tác mà hệ thống hỏi, cảnh báo, thông báo với người sử dụng -> đối thoại, trao đổi giữa hệ thống với người sử dụng.

Câu 29: Trình bày phương pháp thiết kế kiểm thử hộp đen (What) và kiểm thử hộp trắng (How).

Phương pháp hộp đen:

- Chỉ quan tâm đến đầu vào và đầu ra của một đơn vị chương trình, chủ yếu dựa vào đặc tả số liệu nhập/xuất, không quan tâm đến nội dung bên trong của đơn vị đó
 - Phương pháp mang tính ngẫu nhiên vì các lần thử nghiệm chỉ tập trung trong một vùng giới hạn nào đó của không gian thử nghiệm rất có thể có khả năng sai bên ngoài độ ổn định kém
 - Nguyên tắc: Phương pháp hộp đen chọn đặc tả số liệu thử nghiệm rồi phân lớp các số liệu theo số liệu nhập và số liệu kết quả
- Cách chọn các bộ số liệu thử nghiệm:
- Bước 1: Phân lớp không gian thử nghiệm Phân lớp dựa theo:
- tham số nhập

- kết quả xuất

Bước 2: Chọn phần tử đại diện

Phương pháp hộp trắng:

-Chỉ quan tâm đến nội dung bên trong của đơn vị chương trình, chủ yếu dựa vào cấu trúc điều kiện ở bên trong thủ tục

-Phương pháp được thực hiện theo ý niệm “CON ĐƯỜNG THỰC HIỆN CỦA THỦ TỤC”

-Con đường thực hiện của một đơn vị chương trình là mỗi đơn vị chương trình trong một lần thực hiện sẽ có con đường thực hiện riêng, đó là dãy có thứ tự các lệnh bên trong đơn vị chương trình đã được thực hiện trong một lần gọi tương ứng

Câu 30: Liệt kê các phương pháp phân tích thiết kế phần mềm. Trình bày mối quan hệ giữa phân tích thiết kế và đặc tả phần mềm?

Các phương pháp phân tích thiết kế phần mềm:

1. Phương pháp trực tiếp:

- Áp dụng khi thực hiện phần mềm không thông qua giai đoạn phân tích
- Việc thiết kế nhận kết quả chuyển giao trực tiếp từ giai đoạn xác định yêu cầu
- Là quá trình cho phép chuyển đổi từ các yêu cầu (kết quả của giai đoạn xác định yêu cầu) đến mô hình phần mềm tương ứng.
- Mục tiêu chính của thiết kế là mô tả các thành phần của phần mềm tương ứng với các yêu cầu của phần mềm (yêu cầu chức năng nghiệp vụ, yêu cầu chức năng hệ thống, yêu cầu chức năng phi hệ thống)

2. Phương pháp gián tiếp:

- Áp dụng với các quy trình có giai đoạn phân tích
- Việc thiết kế chỉ nhận một phần kết quả chuyển giao trực tiếp từ giai đoạn xác định yêu cầu, phần chính sẽ được nhận gián tiếp qua giai đoạn phân tích. Mô hình phần mềm sẽ được xây dựng tương ứng theo các mô hình trong giai đoạn phân tích.
- Là quá trình cho phép chuyển đổi qua mô hình thế giới thực(kết quả giai đoạn phân tích) đến mô hình phần mềm tương ứng.
- Mục tiêu chính là mô tả các thành phần của phần mềm tương ứng với các mô hình của thế giới thực(mô hình xử lý, mô hình dữ liệu)

Mối quan hệ giữa phân tích thiết kế và đặc tả phần mềm

có mối quan hệ mật thiết với nhau. Quá trình đặc tả phần mềm là nền tảng để thực hiện việc phân tích thiết kế.

Đặc tả yêu cầu phần mềm là tất cả các yêu cầu về phần mềm mà khách hàng- người sử dụng phần mềm nêu ra. Mục đích của yêu cầu phần mềm là xác định được phần mềm đáp ứng được các yêu cầu của khách hàng. Vì vậy mà qua giai đoạn đặc tả yêu cầu phần mềm, người phân tích thiết kế có thể biết được các yêu cầu của người dùng về phần mềm để từ đó đưa ra hướng thiết kế đúng với yêu cầu khách hàng.

Câu 31. Liệt kê các công việc cần thực hiện trong kỹ thuật kiểm thử hộp đen? Nó giúp kiểm tra những nội dung nào của đối tượng kiểm thử? Ưu điểm của kỹ thuật kiểm thử hộp đen là gì?

Các công việc cần thực hiện trong kỹ thuật kiểm thử hộp đen: Kế hoạch test, thiết kế test, tạo testcase, thực hiện test, báo cáo test.

Kế hoạch test: Chỉ ra rõ ràng mục đích và phạm vi của công đoạn test để kiểm tra xem là test bằng approach như thế nào. Điều chỉnh resource thành viên và quyết định cả schedule.

Thiết kế test: Quyết định xem là sẽ sử dụng cái gì cho mục đích và loại test cần được thực hiện trong công đoạn test đó, chức năng đối tượng test, phương pháp test, import và export test. Ngoài ra cũng quyết định cụ thể hơn nguyên liệu cần thiết để thực hiện test hay tiêu chuẩn quyết định thành công/ không thành công.

Tạo testcase: Tạo document ghi trạng thái trước khi bắt đầu test và **kết quả mong đợi** (kết quả chạy đối tượng test theo điều kiện và trình tự thao tác khi thực hiện test sẽ như thế nào) và **cột trạng thái** (cột ghi lại kết quả thao tác của đối tượng test).

Thực hiện test: Vừa xem testcase vừa cho chạy phần mềm thực tế để tiến hành test, sau đó đánh dấu kết quả bằng dấu pass hoặc fail vào cột trạng thái testcase. Trường hợp có testcase khác với kết quả mong đợi thì ghi dấu fail vào cột trạng thái, rồi tạo bản báo cáo lỗi. Trong bản báo cáo lỗi: trình bày nội dung mô tả hiện tượng khác với kết quả mong đợi và hiện tượng đó phát sinh trong trường hợp như thế nào (thao tác, giả nhập, điều kiện,...)

Báo cáo test: Tóm tắt kết quả để báo cáo. Căn cứ vào **các loại dữ liệu** (mục thực hiện, hiệu quả của việc test, công số thực hiện,...) và **dữ liệu lỗi** (số lỗi được tìm ra, số lỗi theo mức độ quan trọng,...) để đánh giá xem có thỏa mãn tiêu chuẩn pass/ fail của test

không? Ngoài ra cũng đề xuất thêm risk có thể sinh ra sau khi release và mục cần bổ sung trong dự án cho giai đoạn tiếp theo.

Nội dung của đối tượng kiểm thử được kiểm tra:

- Chức năng: đủ, đúng
- Giao diện vào, ra: đủ, phù hợp, đúng, tiện lợi
- Cấu trúc, truy cập dữ liệu: thông suốt, đúng đắn
- Thực thi: trôi chảy, kịp thời, chịu lỗi, phục hồi được
- Khởi đầu – kết thúc: Mỗi tiến trình được thực hiện bình thường

Ưu điểm của kỹ thuật kiểm thử hộp đen:

Kiểm thử hộp đen không có liên quan nào tới mã lệnh

- ⇒ Đánh giá khách quan phần mềm, người kiểm thử có thể không hiểu biết về mã lệnh và tìm ra lỗi mà nhóm phát triển không tìm ra

Câu 32. Thiết kế ca kiểm thử trong kỹ thuật kiểm thử hộp trắng cần dựa trên cơ sở nào? Thiết kế ca kiểm thử cần đảm bảo điều kiện gì?

Thiết kế ca kiểm thử trong kỹ thuật kiểm thử hộp trắng cần dựa trên cơ sở:

- Cấu trúc của đơn vị cần kiểm tra
- Các chi tiết thủ tục (Thuật toán)
- Con đường logic (Luồng điều khiển)
- Các trạng thái của chương trình (dữ liệu)

Thiết kế ca kiểm thử cần đảm bảo điều kiện:

- Mọi con đường độc lập trong một modun cần được thực hiện ít nhất một lần
- Mọi ràng buộc logic được thực hiện cả hai phía đúng (true) và sai (false)
- Tất cả các vòng lặp ở biên của nó và các biên vận hành phải được thực hiện
- Mọi cấu trúc dữ liệu nội tại được dùng để bảo đảm hiệu lực thi hành của nó.

Câu 33. Nêu các nội dung chính trong giai đoạn đặc tả phần mềm ? Theo bạn những nội dung nào ảnh hưởng tới chất lượng phần mềm?

Các nội dung chính trong giai đoạn đặc tả phần mềm:

- Phát hiện các yêu cầu phần mềm (Requirements elicitation)
- Phân tích các yêu cầu phần mềm và thương lượng với khách hàng (Requirements analysis and negotiation)

- Mô tả các yêu cầu phần mềm (*Requirements specification*)
- Mô hình hóa hệ thống (*System modeling*)
- Quản trị các yêu cầu phần mềm (*Requirments management*)

Những nội dung ảnh hưởng đến chất lượng phần mềm: Theo em, nội dung quan trọng nhất trong đặc tả phần mềm là phát hiện, phân tích các yêu cầu phần mềm và thương lượng với khách hàng. Vì yêu cầu phần mềm không được phép mơ hồ nếu không sẽ dẫn đến sự hiểu nhầm bởi khách hàng hoặc người phát triển. Với một yêu cầu mơ hồ thì người phát triển sẽ thực hiện nó một cách rẻ nhất còn khách hàng thì không muốn vậy. Do đó khách hàng có thể đòi hỏi sửa đổi chức năng phần mềm khi nó đã gần hoàn thiện khiến cho chi phí tăng và chậm thời điểm bàn giao. Chi phí cho sửa các sai sót trong phát biểu yêu cầu là rất lớn, đặc biệt là khi các sai sót này được phát hiện khi đã bắt đầu xây dựng hệ thống.

Câu 34. Tại sao nói tính linh hoạt và khách quan là yêu cầu quan trọng trong quá trình phân tích và xác định yêu cầu phần mềm.

Tính linh hoạt và khách quan là yêu cầu quan trọng trong quá trình phân tích và xác định yêu cầu phần mềm vì:

- Mục đích của yêu cầu phần mềm là xác định được phần mềm đáp ứng được các yêu cầu và mong muốn của KH
- KH chỉ có những ý tưởng còn mơ hồ về phần mềm cần phải xây dựng để phục vụ công việc của họ, chính vì vậy người phát triển cần kiên trì, nhìn nhận một cách khách quan để xây dựng được một phần mềm có đầy đủ các tính năng cần thiết từ những ý tưởng mơ hồ đó
- KH hay thay đổi các đòi hỏi của mình, người phát triển cần linh hoạt, nắm bắt các thay đổi đó và sửa đổi các mô tả một cách hợp lý.

Câu 35. Các lỗi có thể mắc phải trong quá trình thiết kế phần mềm là gì? Nguyên nhân? Theo bạn lỗi nào là lỗi nặng nhất? Vì sao?

Các lỗi có thể mắc phải trong quá trình thiết kế phần mềm là gì? Nguyên nhân? Theo bạn lỗi nào là lỗi nặng nhất? Vì sao?

Các lỗi có thể mắc trong thiết kế phần mềm gồm:

- a. Lỗi trong thiết kế giao diện

Không chú ý tới thiết kế giao diện cho người dùng

Thường thì các công ty xây dựng website dựa vào quan điểm chủ quan của mình thay vì đứng về góc nhìn của người dùng.

Với lợi ích mà lưu lượng truy cập mang lại, các công ty nên liên tục cập nhật cho trang web của mình để mang lại cho khách hàng những nội dung mà họ muốn, Các doanh nghiệp nên dành thời gian theo dõi các thông số về lượng truy cập để xem trang nào trên website nên được nâng cấp.

Tốc độ load website quá chậm

Tâm lý chung của khách hàng là luôn muốn thu được nhiều thông tin hơn trong một khoảng thời gian nhất định. Hơn thế nữa, google cũng lấy tốc độ tải trang làm một căn cứ để xếp hạng website. Vì vậy, khi thiết kế bạn hãy cố gắng tối ưu chất lượng hình ảnh, nội dung để tốc độ tải trang là nhanh nhất có thể.

- Giao diện web rườm rà, không đồng nhất

Có thể bạn thấy những hiệu ứng nhấp nháy hay hình ảnh pop-up vui nhộn là sáng tạo và hiệu quả, thế nhưng khách hàng của bạn không nghĩ vậy! Họ có xu hướng yêu thích những trang “trông có vẻ chuyên nghiệp” và đồng nhất từ màu sắc cho đến tên thương hiệu. Chính vì vậy một giao diện web được thiết kế với bố cục không rõ ràng, rối rắm sẽ là lỗi thiết kế web thường gặp đầu tiên bạn không nên mắc phải

Nút lệnh quá nhỏ

Các nút lệnh tạo chuyển đổi như đăng kí ngay, mua hàng, đăng nhập... quyết định doanh số sản phẩm. Do đó, bạn nên cân nhắc căn chỉnh kích thước sao cho diện tích nút lệnh phù hợp với chiều rộng ngón tay cái (bởi ngón tay cái là ngón được sử dụng nhiều nhất khi người dùng lướt smart phone). Thêm vào đó, bạn cũng đừng quên để ý tới khoảng cách các nút lệnh để khách hàng không bấm nhầm.

Chưa đi sâu, tìm hiểu, nắm bắt sản phẩm tâm lý và hành vi khách hàng

Việc phân tích hành vi mua sắm, cách tìm hiểu thông tin về sản phẩm và tỉ lệ chuyển đổi ở các dòng điện thoại di động của khách hàng mới chính là yếu tố quyết định. Việc thu thập dữ liệu đầu vào này sẽ cho bạn một đề bài cụ thể về việc **thiết kế phiên bản mobile** cho nhóm đối tượng khách hàng mục tiêu. Điều này sẽ giúp mang lại thành công cho một website doanh nghiệp cũng như công ty của bạn.

b. Lỗi trong thiết kế chương trình

- Chỉ chú ý tới giao diện mà bỏ qua nội dung

Phải cân đối giữa việc thiết kế giao diện và xây dựng nội dung để trang web của bạn được hoàn chỉnh nhất. Không nên đầu tư tất cả về phần giao diện mà bỏ qua việc xây dựng nội dung, vì website của bạn có đẹp đến đâu mà khi người dùng vào trang web không tìm được thông tin hữu ích thì cũng sẽ "vô dụng" mà thôi.

- Khó tìm kiếm thông tin hoặc liên hệ

Thông thường khi khách hàng truy cập vào một website nào đó ngoài việc tìm kiếm nội dung và sản phẩm của doanh nghiệp đó thì điều mà họ tìm kiếm nữa đó chính là thông tin liên hệ của doanh nghiệp. Rất nhiều doanh nghiệp khi thiết kế website lại giấu đi thông tin này, hoặc là để thông tin một nơi mà khách hàng rất khó tìm kiếm.

- Không cập nhật thông tin thường xuyên

Hiện nay có rất nhiều trang web mặc dù được thành lập từ rất lâu đời nhưng thông tin thì rất là cũ. Bởi vì nội dung không được thay đổi theo thời gian khiến nội dung đó đã "lạc hậu" và không còn phù hợp so với thời điểm này nữa.

Do đó một trang web cần được làm tươi mới bằng cách cập nhật thông tin liên tục, nếu thông tin đã lỗi thời thì phải cần được chỉnh sửa lại cho phù hợp và chuẩn xác. Kỹ hơn thì bạn có thể xóa bài viết cũ và viết lại thông tin mới để hướng tới nội dung hữu ích nhất cho người dùng nhé.

- Chưa tạo sự tương tác với khách hàng

Nhiều website thường chú trọng vào việc upload thông tin dịch vụ lên trang web mà quên đi việc phải tạo sự tương tác hai chiều với khách hàng. Chưa tích hợp được ý kiến phản hồi của người dùng vào trang web. Người thiết kế nên đưa thêm chức năng này vào website. Đó như là một bản khảo sát thị trường, thấy được nhu cầu khách hàng đang cần gì để đề ra những chiến lược thích hợp cho công ty.

c. Lỗi thiết kế các tập tin dữ liệu

- **Các dữ liệu chưa đủ và còn trùng lặp**

- Đặt link nhiều lần

Khi click chuột vào link nếu chưa thấy mục người xem cần thì họ bỏ đi ngay. Giả sử như nhiều trang web để có thể down một tài liệu hay một phần mềm nào đó, người dùng phải trải qua 3,4 lần click chuyển trang. Điều này gây phiền toái và mất kiên nhẫn. Vì thế yêu cầu đặt ra khi thiết kế website là tránh đặt link rườm rà.

Nguyên nhân:

- + Chưa đi sâu, tìm hiểu, nắm bắt sản phẩm tâm lý và hành vi khách hàng
- + Kinh nghiệm làm việc chưa đủ
- + Trao đổi thông tin giữa khách hàng và người thiết kế không đầy đủ, chưa hiểu rõ mong muốn của nhau

Lỗi nặng nhất: theo tôi thấy lỗi thiết kế sai yêu cầu của khách hàng là nghiêm trọng nhất vì nếu hiểu sai yêu cầu thì toàn bộ các quá trình sau đó sẽ làm sai toàn bộ cả phần mềm.

Câu 36. Trình bày các giai đoạn để thiết kế một sản phẩm phần mềm? Liệt kê các phương pháp thiết kế phần mềm hiện nay mà bạn biết?

Trình bày các giai đoạn để thiết kế một sản phẩm phần mềm? Liệt kê các phương pháp thiết kế phần mềm hiện nay mà bạn biết?

Quá trình thiết kế

Thiết kế phần mềm là quá trình chuyển các đặc tả yêu cầu dịch vụ thông tin của hệ thống thành đặc tả hệ thống phần mềm. Thiết kế phần mềm trải qua một số giai đoạn chính sau:

Nghiên cứu để hiểu ra vấn đề. Không hiểu rõ vấn đề thì không thể có được thiết kế hữu hiệu.

Mô tả trừu tượng cho mỗi nội dung trong giải pháp. Trước khi tạo ra các tư liệu chính thức người thiết kế cần phải xây dựng một mô tả ban đầu sơ khai rồi chi tiết hóa nó. Các sai sót và khiếm khuyết trong mỗi mức thiết kế trước đó được phát hiện và phải được chỉnh sửa trước khi lập tư liệu thiết kế.

Kết quả của mỗi hoạt động thiết kế là một đặc tả thiết kế. Đặc tả này có thể là một đặc tả trừu tượng, hình thức và được tạo ra để làm rõ các yêu cầu, nó cũng có thể là một đặc tả về một phần nào đó của hệ thống phải được thực hiện như thế nào. Khi quá trình thiết kế tiến triển thì các chi tiết được bổ sung vào đặc tả đó. Các kết quả cuối cùng là các đặc tả về các thuật toán và các cấu trúc dữ liệu được dùng làm cơ sở cho việc thực hiện hệ thống. Các hoạt động thiết kế chính trong một hệ thống phần mềm lớn:

Các nội dung chính của thiết kế là:

- +Thiết kế giao diện gồm thiết kế thủ tục, thiết kế việc thu thập dữ liệu thông qua các biểu mẫu, thiết kế các tài liệu xuất, thiết kế màn hình và menu.

+Thiết kế chương trình mà không viết chương trình cụ thể cho các chức năng: đối ngoại, xử lý lỗi, vào/ra, tra cứu CSDL, Module điều hành, trong các biểu đồ luồng dữ liệu.

+Thiết kế tập tin dữ liệu

Quá trình này được lặp lại cho đến khi các thành phần hợp thành của mỗi hệ con được xác định đều có thể ánh xạ trực tiếp vào các thành phần ngôn ngữ lập trình, chẳng hạn như các gói, các thủ tục và các hàm.

Các phương pháp thiết kế phần mềm hiện nay:

Phương pháp thiết kế phần mềm từ đỉnh xuống Top Down Design: phương pháp này áp dụng để thiết kế phần mềm cho những đơn vị chưa có phần mềm hỗ trợ cho bất kì nghiệp vụ nào, tức là bắt đầu tin học hóa quá trình kinh doanh truyền thống.

Phương pháp thiết kế từ dưới lên Bottom Up Design: phương pháp này áp dụng cho những doanh nghiệp đã áp dụng tin học vào một số bộ phận.

Câu 37. Thế nào là một thiết kế tốt? Chất lượng của thiết kế được thể hiện qua các yếu tố gì? Các yếu tố đó có vai trò như thế nào trong thiết kế phần mềm?

Thế nào là một thiết kế tốt? Chất lượng của thiết kế được thể hiện qua các yếu tố gì?

Các yếu tố đó có vai trò như thế nào trong thiết kế phần mềm?

Không có cách nào hay để xác định được thế nào là thiết kế tốt. Tiêu chuẩn dễ bảo trì là tiêu chuẩn tốt cho người dùng. Một thiết kế dễ bảo trì có thể thích nghi với việc cải biên các chức năng và việc thêm các chức năng mới. Một thiết kế như thế phải dễ hiểu và việc sửa đổi chỉ có hiệu ứng cục bộ. Các thành phần thiết kế phải là kết dính (cohesive) theo nghĩa là tất cả các bộ phận trong thành phần phải có một quan hệ logic chặt chẽ, các thành phần ghép nối (coupling) với nhau là lỏng lẻo. Ghép nối càng lỏng lẻo thì càng dễ thích nghi, nghĩa là càng dễ sửa đổi để phù hợp với hoàn cảnh mới.

Để xem một thiết kế có là tốt hay không, người ta tiến hành thiết lập một số độ đo chất lượng thiết kế:

Sự kết dính (Cohesion) : Sự kết dính của một mô đun là độ đo về tính khớp lại với nhau của các phần trong mô đun đó. Nếu một mô đun chỉ thực hiện một chức năng logic hoặc là một thực thể logic, tức là tất cả các bộ phận của mô đun đó đều tham gia vào việc thực hiện một công việc thì độ kết dính là cao. Nếu một hoặc nhiều bộ phận không tham gia trực tiếp vào việc chức năng logic đó thì mức độ kết dính của

nó là thấp. Thiết kế là tốt khi độ kết dính cao. Khi đó chúng ta sẽ dễ dàng hiểu được từng môđun và việc sửa chữa một môđun sẽ không (ít) ảnh hưởng tới các môđun khác. Constantine và Yourdon định ra 7 mức kết dính theo thứ tự tăng dần sau đây:

Kết dính gom góp: các công việc không liên quan với nhau, song lại bị bó vào một môđun.

Kết dính logic: các thành phần cùng thực hiện các chức năng tương tự về logic chẳng hạn như vào/ra, xử lý lỗi,... được đặt vào cùng một môđun.

Kết dính thời điểm: tất cả các thành phần cùng hoạt hóa một lúc, chẳng hạn như các thao tác khởi tạo được bó lại với nhau.

Kết dính thủ tục: các phần tử trong môđun được ghép lại trong một dãy điều khiển.

Kết dính truyền thông: tất cả các phần tử của môđun cùng thao tác trên một dữ liệu vào và đưa ra cùng một dữ liệu ra.

Kết dính tuần tự: trong một môđun, đầu ra của phần tử này là đầu vào của phần tử khác.

Kết dính chức năng: Mỗi phần của môđun đều là cần thiết để thi hành cùng một chức năng nào đó. Các lớp kết dính này không được định nghĩa chặt chẽ và cũng không phải luôn luôn xác định được. Một đối tượng kết dính nếu nó thể hiện như một thực thể đơn: tất cả các phép toán trên thực thể đó đều nằm trong thực thể đó. Vậy có thể xác định một lớp kết dính nữa là:

Kết dính đối tượng: mỗi phép toán đều liên quan đến thay đổi, kiểm tra và sử dụng thuộc tính của một đối tượng, là cơ sở cung cấp các dịch vụ của đối tượng.

Sự ghép nối (Coupling): Ghép nối là độ đo sự nối ghép với nhau giữa các đơn vị (môđun) của hệ thống. Hệ thống có nối ghép cao thì các môđun phụ thuộc lẫn nhau lớn. Hệ thống nối ghép lỏng lẻo thì các môđun là độc lập hoặc là tương đối độc lập với nhau và chúng ta sẽ dễ bảo trì nó. Các môđun được ghép nối chặt chẽ nếu chúng dùng các biến chung và nếu chúng trao đổi các thông tin điều khiển (ghép nối chung nhau và ghép nối điều khiển). Ghép nối lỏng lẻo đạt được khi bảo đảm rằng các thông tin cục bộ được che dấu trong các môđun và các môđun trao đổi

thông tin thông qua danh sách tham số (giao diện) xác định. Có thể chia ghép nối thành các mức từ chặt chẽ đến lỏng lẻo như sau:

Ghép nối nội dung: hai hay nhiều môđun dùng lẫn dữ liệu của nhau, đây là mức xấu nhất, thường xảy ra đối với các ngôn ngữ mức thấp dùng các dữ liệu toàn cục hay lạm dụng lệnh GOTO.

Ghép nối chung: một số môđun dùng các biến chung, nếu xảy ra lỗi thao tác dữ liệu, sẽ khó xác định được lỗi đó do môđun nào gây ra.

Ghép nối điều khiển: một môđun truyền các thông tin điều khiển để điều khiển hoạt động của một môđun khác.

Ghép nối dư thừa: môđun nhận thông tin thừa không liên quan trực tiếp đến chức năng của nó, điều này sẽ làm giảm khả năng thích nghi của môđun đó.

Ghép nối dữ liệu: Các môđun trao đổi thông tin thông qua tham số và giá trị trả lại.

Ghép nối không có trao đổi thông tin: môđun thực hiện một chức năng độc lập và hoàn toàn không nhận tham số và không có giá trị trả lại.

Ưu việt của thiết kế hướng đối tượng là do bản chất che dấu thông tin của đối tượng dẫn tới việc tạo ra các hệ ghép nối lỏng lẻo. Việc thừa kế trong hệ thống hướng đối tượng lại dẫn tới một dạng khác của ghép nối, ghép nối giữa đối tượng mức cao và đối tượng kế thừa nó.

Sự hiểu được (Understandability): Sự hiểu được của thiết kế liên quan tới một số đặc trưng sau đây:

Tính kết dính: có thể hiểu được thành phần đó mà không cần tham khảo tới một thành phần nào khác hay không?

Đặt tên: phải chăng là mọi tên được dùng trong thành phần đó đều có nghĩa?

Tên có nghĩa là những tên phản ánh tên của thực thể trong thế giới thực được mô hình bởi thành phần đó.

Soạn tư liệu: Thành phần có được soạn thảo tư liệu sao cho ánh xạ giữa các thực thể trong thế giới thực và thành phần đó là rõ ràng.

Độ phức tạp: độ phức tạp của các thuật toán được dùng để thực hiện thành phần đó như thế nào? Độ phức tạp cao ám chỉ nhiều quan hệ giữa các thành phần khác nhau của thành phần thiết kế đó và một cấu trúc logic phức tạp

mà nó dính líu đến độ sâu lồng nhau của cấu trúc ifurtherelse. Các thành phần phức tạp là khó hiểu, vì thế người thiết kế nên làm cho thiết kế thành phần càng đơn giản càng tốt. Đa số công việc về đo chất lượng thiết kế được tập trung vào cố gắng đo độ phức tạp của thành phần và từ đó thu được một vài độ đo về sự dễ hiểu của thành phần. Độ phức tạp phản ánh độ dễ hiểu, nhưng cũng có một số nhân tố khác ảnh hưởng đến độ dễ hiểu, chẳng hạn như tổ chức dữ liệu và kiểu cách mô tả thiết kế. Các số đo độ phức tạp có thể chỉ cung cấp một chỉ số cho độ dễ hiểu của một thành phần.

Sự thích nghi được (Adaptability): Một thiết kế dễ bảo trì thì nó phải sẵn sàng thích nghi được, nghĩa là các thành phần của chúng nên được ghép nối lỏng lẻo. Một thành phần có thể là ghép nối lỏng lẻo theo nghĩa là chỉ hợp tác với các thành phần khác thông qua việc truyền các thông báo. Sự thích nghi được còn có nghĩa là thiết kế phải được soạn thảo tư liệu tốt, dễ hiểu và nhất quán. Để có độ thích nghi thì hệ thống còn cần phải phải tự chứa. Muốn là tự chứa một cách hoàn toàn thì một hệ thống không nên dùng các thành phần khác được xác định ngoại lai. Tuy nhiên, điều đó lại mâu thuẫn với kinh nghiệm nói rằng các thành phần hiện có nên là dùng lại được. Vậy là cần có một cân bằng giữa tính ưu việt của sự dùng lại các thành phần và sự mất mát tính thích nghi được của hệ thống. Một trong những ưu việt chính của kế thừa trong thiết kế hướng đối tượng là các thành phần này có thể sẵn sàng thích nghi được. Cơ cấu thích nghi được này không dựa trên việc cải biên thành phần đã có mà dựa trên việc tạo ra một thành phần mới thừa kế các thuộc tính và các chức năng của thành phần đó. Chúng ta chỉ cần thêm các thuộc tính và chức năng cần thiết cho thành phần mới. Các thành phần khác dựa trên thành phần cơ bản đó sẽ không bị ảnh hưởng gì.

Câu 38. Định nghĩa kiểm thử trong phát triển phần mềm, trình bày những điểm cần lưu ý khi thực hiện kiểm thử.

Định nghĩa kiểm thử trong phát triển phần mềm, trình bày những điểm cần lưu ý khi thực hiện kiểm thử

Định nghĩa: là quá trình khảo sát, xem xét, kiểm tra lại đặc tả, phân tích, thiết kế và mã hóa nhằm phát hiện lỗi của phần mềm, xác minh phần mềm có đúng theo đặc tả, thiết kế, có đáp ứng nhu cầu người dùng, có hoạt động hiệu quả hay không.

Những lưu ý khi kiểm thử:

- +Chất lượng phần mềm do khâu thiết kế quyết định là chủ yếu.
- +Tính dễ kiểm thử phụ thuộc vào cấu trúc chương trình.
- +Người kiểm thử và người xây dựng phần mềm nên khác nhau.
- +Dữ liệu thử cho kết quả bình thường thì không có ý nghĩa nhiều, cần có những dữ liệu kiểm thử phát hiện ra lỗi của phần mềm.
- +Khi thiết kế trường hợp thử, không chỉ dữ liệu kiểm thử nhập vào mà phải thiết kế trước cả dữ kiện kết quả sẽ có.
- +Khi phát sinh thêm trường hợp thử thì nên thử trường hợp thử trước đó để tránh ảnh hưởng lan truyền sóng khi kiểm thử.

Câu 39: So sánh hai phương pháp: kiểm thử trên bàn (kiểm thử tĩnh) và kiểm thử trên máy (kiểm thử động)

giống câu 24

Câu 40: Khái niệm ca kiểm thử? Mục tiêu thiết kế ca kiểm thử? các bước để xây dựng một ca kiểm thử?

Khái niệm ca kiểm thử? Mục tiêu thiết kế ca kiểm thử? các bước để xây dựng một ca kiểm thử?

Khái niệm ca kiểm thử là những trường hợp có thể xảy ra trong quá trình sử dụng phần mềm mà người kiểm thử tạo ra để kiểm tra lỗi của phần mềm.

Mục tiêu thiết kế ca kiểm thử:

- +Tìm ra nhiều lỗi nhất
- + Với nỗ lực và thời gian ngắn nhất.

Các bước xây dựng một ca kiểm thử

Thiết kế test

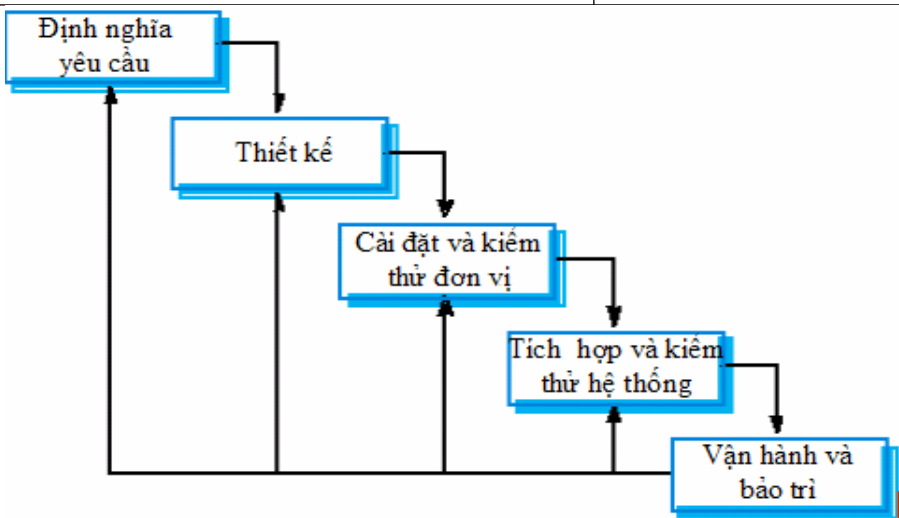
Thiết kế test nhằm chỉ định các test case và các bước kiểm tra chi tiết cho mỗi phiên bản phần mềm.

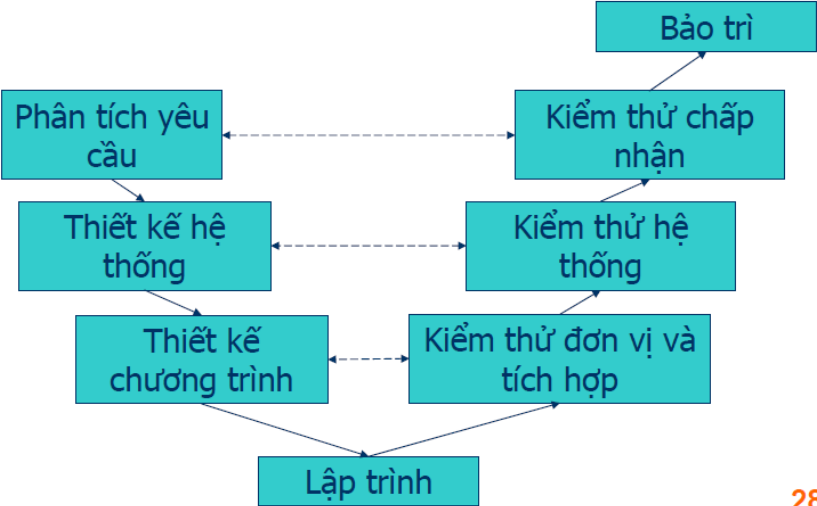
Giai đoạn thiết kế test hết sức quan trọng, nó đảm bảo tất cả các tình huống kiểm tra “quét” hết tất cả yêu cầu cần kiểm tra.

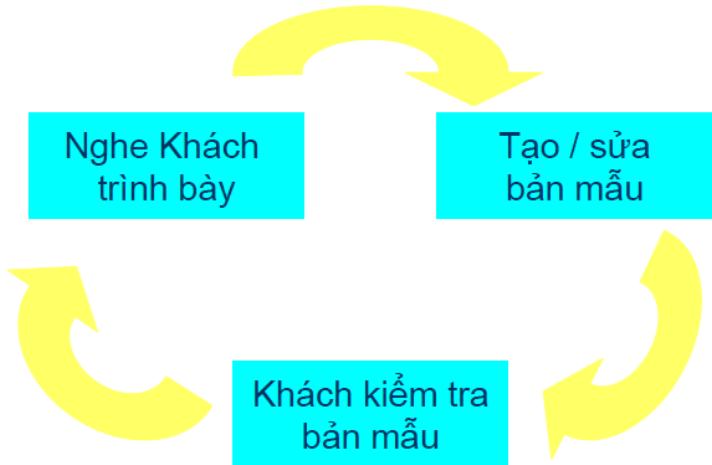
Nó không chỉ làm một lần, nó sẽ được chỉnh sửa, cập nhật, thêm hoặc bớt xuyên suốt chu kỳ phát triển phần mềm, vào bất cứ lúc nào có sự thay đổi yêu cầu hoặc sau khi phân tích thấy cần được sửa chữa hoặc bổ sung.

Các bước thiết kế test:

- Xác định và mô tả test case
- Mô tả các bước chi tiết để kiểm tra
- Xem xét và khảo sát độ bao phủ của việc kiểm tra
- Xem xét test case và các bước kiểm tra

Mô hình	Đặc trưng	Quy mô dự án
Thác nước	 <pre> graph TD A[Định nghĩa yêu cầu] --> B[Thiết kế] B --> C[Cài đặt và kiểm thử đơn vị] C --> D[Tích hợp và kiểm thử hệ thống] D --> E[Vận hành và bảo trì] E --> A </pre>	<ul style="list-style-type: none"> - Là mô hình cổ điển. - Phương pháp áp dụng 1 lần. - Điều khiển hiệu quả. - Phạm vi: giới hạn trong vòng lặp. - Vòng đời (thời gian thực hiện) dài. - Các pha được thực hiện tuần tự, kết quả của pha trước là đầu vào của pha sau. - Ưu: <ul style="list-style-type: none"> • Phù hợp với dự án nhỏ. • Phù hợp với dự án có yêu cầu xác định. - Nhược: <ul style="list-style-type: none"> • Rất khó khăn trong việc thay đổi các pha đã được thực hiện. • Không phù hợp với dự án lớn. • Không phù hợp với mô hình thực tế.
		<ul style="list-style-type: none"> - Không thích hợp với các hệ thống không rõ ràng (mục tiêu từng pha không rõ). - Thích hợp khi các yêu cầu đã được tìm hiểu rõ ràng và những thay đổi sẽ được giới hạn một cách rõ ràng trong suốt quá trình thiết kế. - Phù hợp với dự án nhỏ. - Thời gian thực hiện lâu.

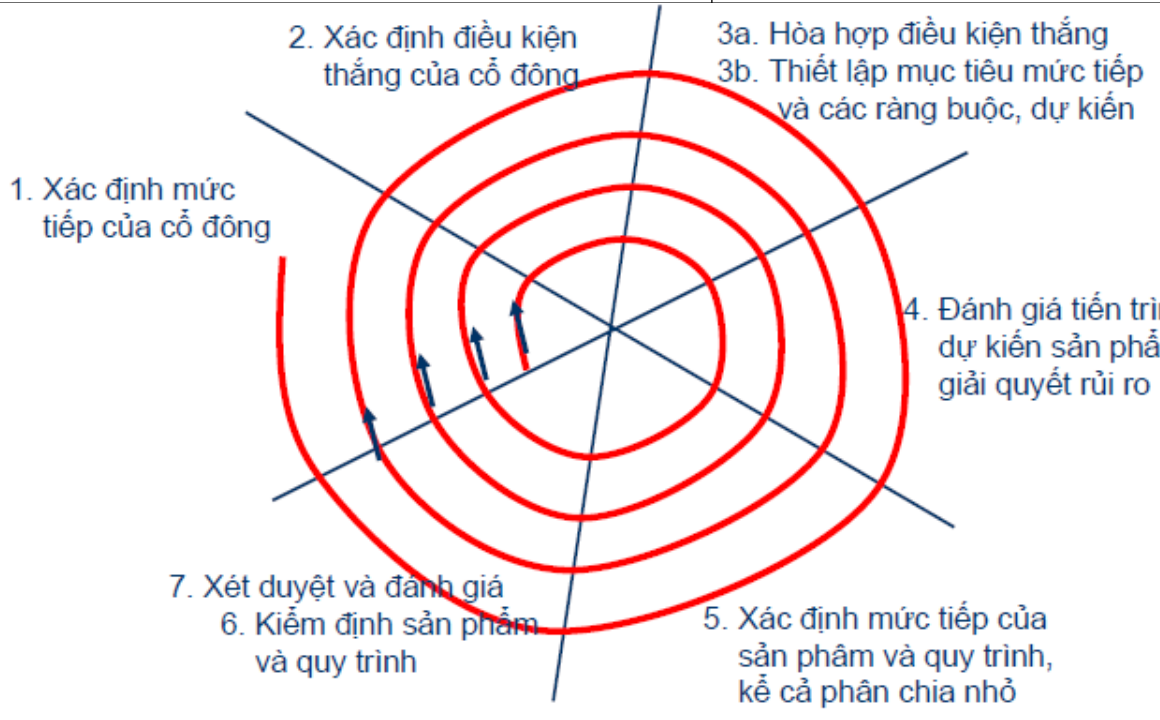
<p>Chữ V</p>	 <p style="text-align: right;">28</p>
	<div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <ul style="list-style-type: none"> - Các tiến trình kiểm thử được thêm vào, tiến hành song song với các pha. - Pha sau chỉ được thực hiện khi pha trước đã thực hiện xong. - Kết nối kiểm thử với phân tích và thiết kế. - Các bước chi tiết hơn mô hình thác nước. - ƯĐ: <ul style="list-style-type: none"> • Sau khi lập trình có thể sửa lỗi luôn do có kiểm thử song song với các phần trước. • Giảm bớt thời gian so với mô hình thác nước. • Có thể đạt được phần mềm có chất lượng, các pha tương thích với nhau, hỗ trợ cho nhau. - NĐ: <ul style="list-style-type: none"> • Nhiều giai đoạn. • Người sd không có cơ hội tham gia trong suốt thời gian của các giai đoạn trung gian từ thiết kế đến kiểm thử. </div> <div style="width: 48%;"> <ul style="list-style-type: none"> - Thích hợp với trường hợp bài toán không nhất quán. - Phù hợp với dự án nhỏ. - Thời gian thực hiện lâu. </div> </div>

Bản mẫu	
	<div> <div> <p>- Bản mẫu là chương trình mẫu có thể thực hiện một phần hay tất cả chức năng mong muốn nhưng ở mức sơ lược.</p> <p>- Tuần tự làm bản mẫu:</p> <p>Tập hợp yêu cầu -> thiết kế nhanh -> xây dựng bản mẫu -> đánh giá của khách hàng-> làm mịn -> quay lại thiết kế nhanh để điều chỉnh -> xây dựng sản phẩm.</p> <p>- Mô hình bản mẫu thường được sd</p> <ul style="list-style-type: none"> • khi mới rõ mục đích chung của PM, chưa rõ chi tiết đầu vào hay xử lý ra sao hoặc chưa rõ yêu cầu đầu ra. • khi các giải thuật, kỹ thuật dùng làm bản mẫu có thể chưa nhanh, chưa tốt, miễn là có mẫu để thảo luận gợi ý/c của người dùng. • như “hệ sơ khai” để thu thập y/c người dùng qua các thiết kế nhanh. <p>- ƯĐ:</p> <ul style="list-style-type: none"> • Theo kịp sự thay đổi của khách hàng. • Giảm thiểu rủi ro. • Phù hợp với hệ thống rủi ro cao, chiến lược cài đặt chưa rõ ràng, yêu cầu không chắc </div> <div> <p>- Thích hợp với bài toán bé, yêu cầu không chắc chắn, thời gian chậm.</p> </div> </div>

	<p>chẩn.</p> <p>- NĐ:</p> <ul style="list-style-type: none"> Khách hàng có thể cho rằng bản mẫu là hệ thống thực, tuy nhiên bản mẫu có thể thay đổi trong quá trình xây dựng và thiết kế -> mong đợi của khách hàng có thể không thực tế. Người phát triển có sự lựa chọn không tốt, có thể phù hợp với nguyên mẫu nhưng không phù hợp với hệ thống thực. Nguyên mẫu không giống hoàn toàn với hệ thống cuối cùng, dẫn đến phản ứng của KH có thể k tốt. <p>Do sự hoàn thiện dần nên nhiều khi tính cấu trúc không cao -> khó kiểm soát, bảo trì.</p>	
Phát triển ứng dụng nhanh	<p>60 - 90 days</p>	<p>- là mô hình phát triển PM gia tăng, tăng dần từng bước với mỗi chu trình phát triển rất ngắn (60 – 90 ngày).</p> <p>- Quy mô dự án lớn, thời gian nhanh, y/c rõ ràng. (tiếp cột bên)</p>

	<ul style="list-style-type: none"> - xây dựng dựa trên hướng thành phần với khả năng tái sd. - gồm 1 số nhóm, mỗi nhóm làm 1 RAD theo các pha: mô hình nghiệp vụ, mô hình dữ liệu, mô hình xử lý, tạo ứng dụng, kiểm thử và đánh giá. - Mô hình nghiệp vụ (business modeling): luồng thông tin được mô hình hóa để trả lời các câu hỏi: <ul style="list-style-type: none"> • Thông tin nào điều khiển xử lý các nghiệp vụ? Thông tin gì được sinh ra? Ai sinh ra nó? Thông tin đi đến đâu? Ai xử lý chúng? - Mô hình tiến trình và dữ liệu: <ul style="list-style-type: none"> • Data modeling: các đối tượng dữ liệu cần để hỗ trợ nghiệp vụ (business). Định nghĩa các thuộc tính của từng đối tượng và xác lập quan hệ giữa các đối tượng. • Process modeling: Các đối tượng dữ liệu được chuyển sang luồng thông tin thực hiện chức năng nghiệp vụ. Tạo mô tả xử lý để cập nhật (thêm, sửa, xóa, khôi phục) từng đối tượng dữ liệu - Tự sinh ứng dụng và kiểm thử: <ul style="list-style-type: none"> • Application Generation: Dùng các kỹ thuật thể hệ 4 để tạo phần mềm từ các thành phần có sẵn hoặc tạo ra các thành phần có thể tái sử dụng lại sau này. Dùng các công cụ tự động để xây dựng phần mềm • Testing and Turnover: Kiểm thử các thành phần mới và kiểm chứng mọi giao diện (các thành phần cũ đã được kiểm thử và dùng lại) 	<ul style="list-style-type: none"> - ƯĐ: <ul style="list-style-type: none"> • Dễ thực hiện đối với các ứng dụng có thể môđun hóa hoặc đòi hỏi tính năng cao. • Theo kịp sự thay đổi của khách hàng. • Giảm thiểu rủi ro. - NĐ: <ul style="list-style-type: none"> • Cần nguồn nhân lực dồi dào để tạo các nhóm cho các chức năng chính. • Yêu cầu hai bên phải có trách nhiệm cao. • RAD không phải tốt cho mọi ứng dụng, nhất là với ứng dụng không thể môđun hóa hoặc đòi hỏi tính năng cao. • Mạo hiểm kỹ thuật cao thì không nên dùng RAD.
--	--	---

<p>Xoắn ốc</p>	<p style="text-align: center;">Spiral</p> <p style="text-align: center;">Boehm 1987</p>
<p>- Quy trình phát triển PM được biểu diễn như 1 vòng xoắn ốc.</p> <p>- Các pha trong quy trình phát triển gồm:</p> <p>Thiết lập mục tiêu.</p> <p>Đánh giá và giảm thiểu rủi ro.</p> <p>Phát triển và đánh giá.</p>	<p>- Mô hình xoắn ốc phù hợp với</p> <ul style="list-style-type: none"> • Các hệ phần mềm quy mô lớn, các dự án lớn, phức tạp • Hệ thống cần phát triển

	<p>Lập kế hoạch.</p> <ul style="list-style-type: none"> - Nhấn mạnh việc đánh giá các rủi ro. - Phần mềm được xây dựng theo nhiều chu kỳ. Mỗi chu kỳ tương ứng với 1 sp của 1 giai đoạn ptien. - Ưu điểm: <ul style="list-style-type: none"> • Hạn chế rủi ro sớm • Nhận được phản hồi (<i>feedbacks</i>) từ khách hàng sớm • Dễ kiểm soát các mạo hiểm ở từng mức tiến hóa - Nhược: <ul style="list-style-type: none"> • Khó thuyết phục khách hàng là phương pháp tiến hóa xoắn ốc có thể kiểm soát được • Chưa được dùng rộng rãi như các mô hình tuyến tính hoặc chế thử. 	<p>nhiều phiên bản</p> <ul style="list-style-type: none"> • Các hệ thống có yêu cầu chưa xác định rõ ràng.
<p>Xoắn ốc WINWIN</p>	 <p>1. Xác định mức tiếp của cổ đông</p> <p>2. Xác định điều kiện thắng của cổ đông</p> <p>3a. Hòa hợp điều kiện thắng 3b. Thiết lập mục tiêu mức tiếp và các ràng buộc, dự kiến</p> <p>4. Đánh giá tiến trình dự kiến sản phẩm giải quyết rủi ro</p> <p>5. Xác định mức tiếp của sản phẩm và quy trình, kể cả phân chia nhỏ</p> <p>6. Kiểm định sản phẩm và quy trình</p> <p>7. Xét duyệt và đánh giá</p>	
	<p>- Là mô hình xoắn ốc nhằm thỏa hiệp giữa người phát triển và khách hàng, cả hai</p>	<p>- Mô hình xoắn ốc WINWIN phù hợp với</p> <ul style="list-style-type: none"> • Các hệ phần mềm quy mô

	<p>cùng “Thắng”(win-win):</p> <ul style="list-style-type: none"> • Khách thì có phần mềm thỏa mãn yêu cầu chính • Người phát triển thì có kinh phí thỏa đáng và thời gian hợp lý <p>- Các hoạt động chính trong việc xác định hệ thống:</p> <ul style="list-style-type: none"> • Xác định cổ đông (stakeholders) • Xác định điều kiện thắng của cổ đông • Thỏa hiệp điều kiện thắng của các bên liên quan <p>- Ưu điểm:</p> <ul style="list-style-type: none"> • Hạn chế rủi ro sớm • Nhận được phản hồi (<i>feedbacks</i>) từ khách hàng sớm • Dễ kiểm soát các mạo hiểm ở từng mức tiến hóa <p>- Nhược:</p> <ul style="list-style-type: none"> • Khó thuyết phục khách hàng là phương pháp tiến hóa xoắn ốc có thể kiểm soát được <p>Chưa được dùng rộng rãi như các mô hình tuyến tính hoặc chế thử.</p>	<p>lớn, các dự án lớn, phức tạp</p> <ul style="list-style-type: none"> • Hệ thống cần phát triển nhiều phiên bản • Các hệ thống có yêu cầu chưa xác định rõ ràng.
--	--	---

Đề 1:

Câu 1: sự phát triển của phần mềm: gồm 3 giai đoạn

❖ Giai đoạn thứ nhất: (~ 1960):

- phần cứng thay đổi liên tục, theo tính chuyên dụng
- xử lý theo lô
- môi trường lập trình có tính cá nhân
- sản xuất đơn chiếc
- >ngôn ngữ: mã máy, đặc thù cho từng máy

->lập trình: là 1 nghệ thuật theo bản năng, chưa có phương pháp

❖ Giai đoạn 2: (~ giữa thập kỷ 70)

- Phần mềm đa nhiệm, đa người sử dụng

- Hệ thống thời gian thực

- Xuất hiện lưu trữ trực tuyến

-> phần mềm trở nên phức tạp dẫn đến nhiều dự án thất bại

->chi phí tăng cao dẫn đến nhu cầu tăng cao

Ngôn ngữ: có cấu trúc: PL1, Algos60, Fortran, COBOL

Lập trình: có phương pháp lập trình

Phát triển hệ thống: chưa có phương pháp, kinh nghiệm là chính

❖ Giai đoạn 3: (~ đến nay)

- Hệ thống phân tán

- Mạng cục bộ, toàn cầu

- Bộ vi xử lý phát triển mạnh

->phần mềm ngày càng phức tạp

->sử dụng nhiều máy tính cá nhân, phần mềm nhúng

->số người dùng tăng nhanh

Ngôn ngữ: ngôn ngữ bậc cao, hướng đối tượng

Phát triển hệ thống: có phương pháp, công cụ tự động

Tiêu chuẩn đánh giá 1 phần mềm tốt: là phần mềm có chất lượng thỏa mãn:

Những yếu tố bên ngoài của 1 sản phẩm như: tốc độ nhanh, chạy ổn định, dễ sử dụng, dễ thích nghi với những thay đổi và một số yếu tố như: tính đúng đắn, tính bền vững, tính có thể mở rộng đc, tính sử dụng lại, tính tương thích , tính hiệu quả, tính bảo trì đc, tính tin cậy...

Những yếu tố chất lượng bên trong như tính dễ đọc, dễ hiểu, sản phẩm trực quan sinh động...

Câu 3: hệ thống thông tin quản lý cửa hàng bán linh kiện máy tính: khi khách hàng yêu cầu mua hàng. bộ phận kế toán có trách nhiệm kiểm tra hàng, nếu còn hàng lập hóa đơn. hóa đơn gửi cho bộ phận thủ kho. thủ kho có trách nhiệm xuất hàng bán cho khách. cuối tháng báo cáo về tình hình hàng tồn kho, công nợ ... cho ban giám

đốc

yêu cầu:

1: nên sử dụng mô hình nào để xây dựng phần mềm. giải thích.

2: yêu cầu chức năng và phi chức năng của hệ thống

3: hệ thống có những thực thể và thuộc tính cơ bản nào!

1. nên sử dụng mô hình chữ V để xây dựng phần mềm

- Lý do lựa chọn mô hình chữ V
 - Bài toán là việc “ quản lý bán linh kiện máy tính của một cửa hàng”, quy mô áp dụng :
 - Mang tính đặc trưng cho cửa hàng (bán linh kiện), không nhất quán.
 - Quy mô không lớn.
 - Chương trình đơn giản, các chức năng đưa ra cho phần mềm cần rõ ràng, dễ sử dụng đối với người sử dụng.
 - Cửa hàng có quy mô vừa và nhỏ, kiểm tra kiểm thử diễn ra thường xuyên. Nội dung công việc gần như là rõ ràng nên phù hợp lựa chọn mô hình chữ V. Đó là một trong các yếu tố lựa chọn mô hình này
 - Trong mô hình chữ V các hoạt động kiểm thử diễn ra song song với các hoạt động liên quan nên có thể nhanh chóng phát hiện ra sai sót để có thể quay lại sửa chữa ngay rất thích hợp với đối tượng mới lập trình và ít có kinh nghiệm
 - Mô hình chữ V kết nối kiểm thử với phân tích và thiết kế nên có thể tránh được những lỗi và có thể dễ dàng kiểm soát lỗi hơn-điều này có ý nghĩa hết sức quan trọng.

2. yêu cầu chức năng và phi chức năng

* Yêu cầu chức năng.

Bao gồm các chức năng chính:

- Nhập và lưu phiếu nhập hàng.
- Nhập và in hoá đơn cho khách hàng tới mua hàng.
- Cập nhập các loại hàng mới và số lượng hiện tại của các loại hàng.
- Thống kê, lập báo cáo hàng tồn, hàng đã bán.

Người sử dụng có thể tìm kiếm tất cả CSDL hoặc 1 tập con

Hệ thống sẽ cung cấp những giao diện thích hợp để ng dùng sử dụng dễ dàng

* Yêu cầu phi chức năng.

❖ Yêu cầu về sản phẩm:

- Hệ thống quản lí bán hàng phải có khả năng sử dụng nhanh, được cài đặt bằng C++ hoặc C#.
- Có cả ngôn ngữ tiếng việt và tiếng anh.
- Hệ thống dễ dàng thay đổi và mở rộng khi cần thiết.

❖ Yêu cầu về hệ thống :

- Hệ thống phải tương thích với Window XP hoặc Window 7, 8.
- Quy trình xây dựng hệ thống và các tài liệu chuyển giao phải thoả mãn các quy tắc của tài liệu hệ thống quản lí bán hàng.
- Quản lý cơ sở dữ liệu (CSDL) bằng Access 2007 hoặc SQL server.
- Có khả năng mở rộng nâng cấp (ví dụ: thêm loại thông tin mới trong CSDL, thêm chức năng quét mã vạch cho hệ thống...).
- Tương thích với các thiết bị hỗ trợ (ví dụ: máy in, máy Fax...).

3. hệ thống gồm những thực thể và thuộc tính

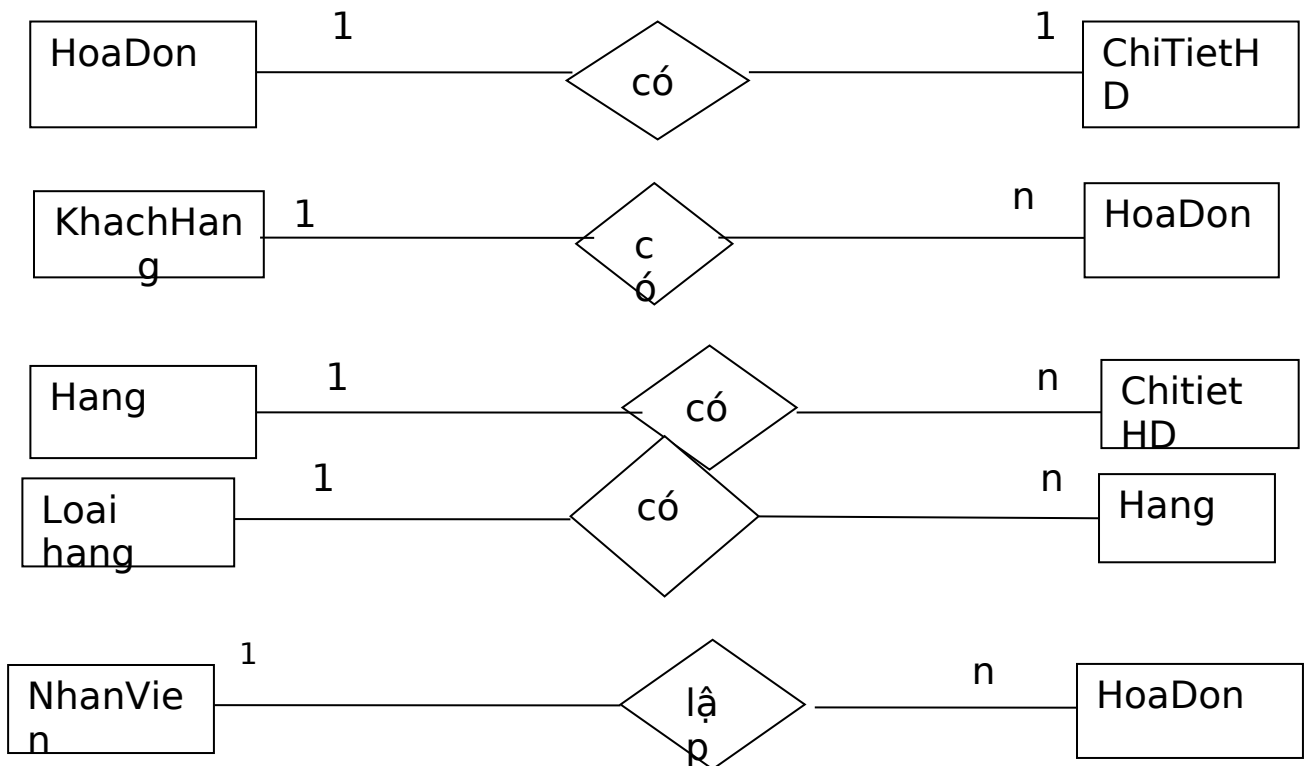
Khách hang(Ma KH, ten, dchi,sdt)

Loai hang(Ma loai, ten loai)

Hang(Ma hang, ten hang, so luong)

Nhan vien(Ma NV, ten NV,ngsinh,gtnh,que quan, dchi, sdt)

Hoa don(Ma HD, ngày lap HD, Gia tri HD)



đề 2:

câu 1:khái niệm vòng đời phần mềm

Là khoảng thời gian tính từ khi phần mềm được đề xuất cho đến khi bỏ đi: cụ thể là từ khi được đặt hàng, phát triển, sử dụng đến khi bị loại bỏ.

❖ các giai đoạn: phân tích và đặc tả yêu cầu, thiết kế phần mềm, lập trình.

kiểm thử,cài đặt, bảo trì

❖ đặc điểm và các hoạt động chính của mỗi giai đoạn:

- phân tích và đặc tả yêu cầu: là bản đặc tả các dịch vụ mà hệ thống cung cấp, các ràng buộc để xây dựng vận hành hệ thống, là khâu đầu tiên trong quá trình xây dựng phần mềm

- Hoạt động chính của giai đoạn này là:

- + Phát hiện yêu cầu
- + Phân tích yêu cầu và thương lượng với khách hàng
- + Mô tả các yêu cầu
- + Mô tả hệ thống
- + Kiểm tra tính hợp lý của yêu cầu
- + Quản trị các yêu cầu

- Thiết kế phần mềm: là thiết kế cấu hình phần cứng và cấu trúc phần mềm để có đc hệ thống thỏa mãn yêu cầu đặt ra
 - Đặc điểm: chọn chiến lược cài đặt và quản lý dữ liệu
 - + Tìm ra nguồn tài nguyên chung và cơ chế điều khiển truy nhập chung
 - + Thiết kế cơ chế điều khiển thích hợp cho hệ thống, kể cả quản lý nhiệm vụ
 - + Xem xét các điều kiện ràng buộc để xử lý ntnao
 - Hoạt động chính của giai đoạn này là:
 - + Thiết kế giao diện
 - + Thiết kế chương trình
 - + Thiết kế tập tin dữ liệu
 - Lập trình: là quá trình chuyển đổi từ thiết kế chi tiết sang mã lệnh. Lựa chọn ngôn ngữ lập trình phụ thuộc vào cấu hình máy, số lượng ngôn ngữ lập trình có sẵn, thói quen sử dụng ngôn ngữ lập trình. Người lập trình cần xác định thông tin tối thiểu cho 1 module chương trình
 - Kiểm thử: là 1 trong những giai đoạn quan trọng nhất phát triển phần mềm, là mẫu chốt của đảm bảo chất lượng phần mềm. là tiến trình xem xét lại đặc tả, thiết kế, mã hóa nhằm phát hiện lỗi phần mềm. kiểm thử thành công khi phát hiện ra lỗi, kiểm thử ko phát hiện ra lỗi là kiểm thử dở
- Một phép kiểm thử bao gồm: tên module kiểm thử, dữ liệu vào, dữ liệu r among muốn, dữ liệu ra thực tế, đánh giá chất lượng phần mềm qua 2 kết quả mong muốn và thực tế
- Cài đặt: lập kế hoạch cài đặt: chuyển đổi phần cứng, CSDL, công nghệ quản lý, phong cách làm việc...biến đổi dữ liệu từ hệ thống cũ sang mới, biên soạn tài liệu hệ thống
 - Bảo trì: là pha cuối cùng của vòng đời gồm 2 hoạt động là quản lý hoạt động bảo trì và chuẩn hóa hoạt động bảo trì

Câu 2: nêu các nguyên tắc thiết kế giao diện

- Các nguyên tắc thiết kế giao diện
 - Sự quen thuộc của người sử dụng: giao diện phải được xây dựng dựa trên các thuật ngữ và các khái niệm mà người sử dụng có thể hiểu được hơn là những khái niệm liên quan đến máy tính. Ví dụ: hệ thống văn

phòng nên sử dụng các khái niệm như thư, tài liệu, cặp giấy ... mà không nên sử dụng những khái niệm như thư mục, danh mục ...

- Thống nhất: hệ thống nên hiển thị ở mức thống nhất thích hợp. Ví dụ: các câu lệnh và menu nên có cùng định dạng ...
- Tối thiểu hoá sự bất ngờ: nếu một yêu cầu được xử lý theo cách đã biết trước thì người sử dụng có thể dự đoán các thao tác của những yêu cầu tương tự.
- Khả năng phục hồi: hệ thống nên cung cấp một số khả năng phục hồi từ lỗi của người sử dụng và cho phép người sử dụng khôi phục lại từ chỗ bị lỗi. Khả năng này bao gồm cho phép làm lại, hỏi lại những hành động như xoá, huỷ ...
- Hướng dẫn người sử dụng: như hệ thống trợ giúp, hướng dẫn trực tuyến ...
- Tính đa dạng: hỗ trợ nhiều loại tương tác cho nhiều loại người sử dụng khác nhau. Ví dụ: nên hiển thị phông chữ lớn với những người cận thị

Nguyên tắc quan trọng nhất là sự quen thuộc của người sử dụng.

Một phần mềm được tạo ra, mục đích hướng tới là cho tất cả đối tượng người sử dụng đều dùng được một cách dễ dàng, đơn giản và dễ hiểu nhất thì đó mới là một phần mềm thành công. Khi đó giao diện phần mềm là cái ảnh hưởng đầu tiên đến khách hàng. Giao diện có thân thiện, dễ hiểu thì người dùng mới có thể sử dụng được các chức năng bên trong của phần mềm. Nếu ngay từ đầu giao diện đã khó hiểu thì người dùng sẽ không biết các chức năng cần thiết ở đâu để dùng. Từ đó họ sẽ k dùng phần mềm đó, dẫn đến thất bại.

Vì vậy sự quen thuộc của người dùng là nguyên tắc quan trọng nhất.

Câu 3 : Hệ thống quản lý bãi xe

Khi xe vào, người trông xe sẽ đối chiếu với bảng phân loại xe, tìm xem có còn chỗ trống trong bãi đỗ xe hay không. Nếu còn chỗ trống cho loại xe đó thì ghi thông tin vào vé xe và cho xe vào gửi đồng thời lưu thông tin vào hệ thống. Nếu không còn chỗ trống thì thông báo để khách hàng tìm bãi đỗ xe khác. Khi khách hàng trả xe, người trông coi sẽ đối chiếu thông tin ghi trên vé xe, nếu phát hiện vé giả hoặc thông tin sai

thì không cho lấy xe. Nếu không có gì sai thì thu tiền của khách và lưu thông tin vào hệ thống

a) Mô hình phát triển PM dùng cho hệ thống trên :

b) Sơ đồ mức đỉnh :

c) Thiết kế các hoạt động kiểm thử cho chức năng « thanh toán » :

Test case : thanh toán

Mô tả	Các bước thực hiện	Kết quả mong đợi	Kết quả thực hiện
Vé đúng, xe đúng			

1. yêu cầu chức năng và phi chức năng, miền ứng dụng

❖ **Yêu cầu chức năng.**

- Bao gồm các chức năng chính:.....
- Người sử dụng có thể tìm kiếm tất cả CSDL hoặc 1 tập con
- Hệ thống sẽ cung cấp những giao diện thích hợp để ng dùng sử dụng dễ dàng

❖ **Yêu cầu phi chức năng.**

- Yêu cầu về sản phẩm:
 - + Hệ thống quản lí phải có khả năng sử dụng nhanh, được cài đặt bằng C++ hoặc C#.
 - + Có cả ngôn ngữ tiếng việt và tiếng anh.
 - + Hệ thống dễ dàng thay đổi và mở rộng khi cần thiết.
- Yêu cầu về hệ thống :
 - + Hệ thống phải tương thích với Window XP hoặc Window 7, 8, 10.
 - + Quy trình xây dựng hệ thống và các tài liệu chuyển giao phải thoả mãn các quy tắc của tài liệu hệ thống quản lí bán hàng.
 - + Quản lý cơ sở dữ liệu (CSDL) bằng Access 2007 hoặc SQL server.
 - + Có khả năng mở rộng nâng cấp (ví dụ: thêm loại thông tin mới trong CSDL, thêm chức năng quét mã vạch cho hệ thống...).

+Tương thích với các thiết bị hỗ trợ (ví dụ: máy in, máy Fax...).

❖ **Yêu cầu miền ứng dụng**

- Giao diện người dùng chuẩn cho tất cả các CSDL dựa trên chuẩn phù hợp với mô hình mạng client/server
- Phụ thuộc vào yêu cầu của người sử dụng, những tài liệu có thể được in ngay trên server và chuyển đến cho người sử dụng hoặc gửi cho máy in mạng

• **Phần mềm quản lý thiết bị máy tính**

- Nhập thiết bị: nhân viên cửa hàng sẽ nhận thiết bị và điền thông tin vào phiếu nhập
- Xuất hàng: nhân viên nhập phiếu xuất. một bản giao cho khách hàng và một bản lưu trữ lại
- Tra cứu thông tin: Nhân viên có thể tra cứu nội dung thông tin của thiết bị, việc nhập và xuất thiết bị.
- Cập nhật thông tin: Cửa hàng cập nhật và in ra những bản báo giá để khách hàng biết thông tin về thiết bị
- Thống kê báo cáo: Dựa vào phiếu xuất nhập có thể tổng kết tình hình bán thiết bị của cửa hàng theo từng ngày, tháng, quý.

• **Quản lý thư viện**

- Quản lý sách: thêm, sửa, xóa , tìm kiếm, tình trạng sách
- Quản lý bạn đọc : thêm sửa xóa tìm kiếm
- Quản lý mượn trả sách: quản lý mượn sách, quản lý trả sách

• **Quản lý giới thiệu việc làm**

- Theo dõi yêu cầu lao động: cập nhật thông tin CV, hiệu chỉnh thông tin
- Theo dõi nhu cầu tìm việc: cập nhật thông tin người LĐ, sửa, xóa tt NLĐ
- Xử lý các yêu cầu: tìm kiếm yêu cầu LĐ, tìm kiếm nhu cầu tìm việc, sắp xếp các vị trí, thanh toán phí dịch vụ
- Lưu trữ hồ sơ
- Tác nhân: nhà tuyển dụng, người lao động, ban quản lý

• **Quản lý khách sạn**

- Quản lý khách hàng
- Quản lý trạng thái phòng
- Quản lý giao dịch: thuê phòng, đặt phòng trước, thanh toán trả phòng, xuất hóa đơn

- Tác nhân: nhân viên, khách hàng
 - **Quản lý bãi xe**
- Nhận xe: nhận dạng xe, kiểm tra chỗ trống, ghi vé, ghi số vào xe
- Trả xe: kiểm tra vé, đối chiếu vé- xe, thanh toán vé, ghi số xe ra
- Giải quyết sự cố: kiểm tra số gửi, kiểm tra hiện trường, lập biên bản, thanh toán
- Báo cáo
- Tác nhân: khách, ban lãnh đạo
 - **Quản lý giải vô địch quốc gia**
- Quản lý thông tin đội bóng
- Quản lý lịch thi đấu, thông tin quy định
- Quản lý bảng xếp hạng
- Báo cáo thống kê trao giải
 - **Quản lý đăng ký học phân**
- Quản lý đăng ký học
- Quản lý đăng ký dạy học
- Quản lý điếm: xem, nhập , sửa
 - **Quản lý hoạt động đoàn trường**
- Quản lý đoàn viên: thêm, sửa, xóa
- Quản lý hoạt động: thu đoàn phí, đánh giá đoàn viên
- Thống kê: đoàn viên chưa đóng đoàn phí, xếp loại