



[123doc] -

250-cau-hoi-trac-nghiem-cong-nghe-phan-mem-co-dap-an

Tư tưởng hồ chí minh (Trường Đại học Bách khoa - Đại học Quốc gia Thành phố Hồ Chí Minh)



Scan to open on Studocu

Câu hỏi:

1/. Tiêu chuẩn ISO-14598 đưa ra:

A/. Đưa ra quy trình đánh giá tính an toàn cho sản phẩm phần mềm.

B/. Đưa ra quy trình đánh giá hiệu quả của phần mềm.

C/. Đưa ra quy trình đánh giá chất lượng cho sản phẩm phần mềm. (đ)

D/. Đưa ra quy trình đánh giá tính khả dụng cho sản phẩm phần mềm.

2/. Trong phát triển phần mềm, yếu tố nào quan trọng nhất?

A/. Con người. (đ)

B/. Quy trình.

C/. Sản phẩm.

D/. Thời gian.

3/. Kỹ thuật nào sau đây là xây dựng phần mềm từ các thành phần đã được thiết kế trong lĩnh vực công nghệ khác nhau?

A/. Extreme programming.

B/. Evolutionary prototyping.

C/. Component architecture. (đ)

D/. Open-source development

4/. IEEE 830-1993 là một khuyến nghị tiêu chuẩn cho

A/. Software requirement specification. (đ)

B/. Software design.

C/. Testing.

D/. Coding.

5/. Kỹ sư phần mềm không cần

A/. Kiến thức về phân tích thiết kế hệ thống.

B/. Kiến thức về cơ sở dữ liệu.

C/. Lập trình thành thạo bằng một ngôn ngữ lập trình. (đ)

D/. Kinh nghiệm quản lý dự án phần mềm.

6/. Tính khả thi của phần mềm dựa vào các yếu tố sau:

A/. Nghiệp vụ và tiếp thị.

B/. Phạm vi, ràng buộc và thị trường.

C/. Công nghệ, tiền bạc, thời gian và tài nguyên. (đ)

D/. Kỹ năng và năng lực của nhà phát triển.

7/. Phần mềm dự báo thời tiết thu thập các số liệu về nhiệt độ, độ ẩm, ... xử lý tính toán để cho ra các dự báo thời tiết là 1 ví dụ của loại phần mềm:

A/. Phần mềm hệ thống (System software)

B/. Phần mềm trí tuệ nhân tạo (Artificial Intelligence Software)

C/. Phần mềm thời gian thực (Real time software) (đ)

D/. Phần mềm nghiệp vụ (Business software)

8/. Loại phần mềm gì là 1 tập hợp các chương trình để cung cấp dịch vụ cho các

chương trình khác:

- A/. Phần mềm hệ thống (System software)
 - B/. Phần mềm trí tuệ nhân tạo (Artificial Intelligence Software)
 - C/. Phần mềm thời gian thực (Real time software) (đ)**
 - D/. Phần mềm nghiệp vụ (Business software)
- 9/. Phần mềm quản lý sinh viên của 1 trường là:
- A/. Phần mềm hệ thống (System software)
 - B/. Phần mềm trí tuệ nhân tạo (Artificial Intelligence Software)
 - C/. Phần mềm thời gian thực (Real time software)
 - D/. Phần mềm nghiệp vụ (Business software)**
- 10/. Phần mềm quản lý tài chính của một công ty là:
- A/. Phần mềm nghiệp vụ (Business software)**
 - B/. Phần mềm hệ thống (System software)
 - C/. Phần mềm trí tuệ nhân tạo (Artificial Intelligence Software)
 - D/. Phần mềm thời gian thực (Real time software)
- 11/. Theo một báo cáo của IBM, "31% các dự án bị hủy bỏ trước khi chúng được hoàn thành, 53% vượt dự toán trung bình 189% và cứ mỗi 100 dự án, có 94 dự án khởi động lại". Lý do nào cho số liệu thống kê trên?
- A/. Thiếu đào tạo đầy đủ về công nghệ phần mềm.**
 - B/. Thiếu đạo đức phần mềm và sự hiểu biết.
 - C/. Quản lý các vấn đề trong công ty.
 - D/. Ảnh hưởng của sự suy thoái kinh tế.
- 12/. Điều nào không đúng?
- A/. Công nghệ phần mềm thuộc ngành khoa học máy tính.
 - B/. Công nghệ phần mềm là một phần của ngành kỹ thuật hệ thống (System Engineering).
 - C/. Khoa học máy tính thuộc ngành công nghệ phần mềm.**
 - D/. Công nghệ phần mềm có liên quan với việc phát triển và cung cấp các phần mềm hữu ích.
- 13/. Mối quan tâm chính của công nghệ phần mềm là gì?
- A/. Sản xuất phần cứng.
 - B/. Sản xuất phần mềm. (đ)**
 - C/. Cấu hình mạng.
 - D/. Phần mềm có thể dùng lại.
- 14/. Điều nào là đặc trưng của một thiết kế phần mềm tốt?
- A/. Thể hiện kết nối mạnh mẽ giữa các mô-đun của nó.
 - B/. Thực hiện tất cả các yêu cầu trong mô hình phân tích. (đ)**
 - C/. Bao gồm các trường hợp thử nghiệm cho tất cả các thành phần
 - D/. Cung cấp một bức tranh hoàn chỉnh của phần mềm.

1/. Theo thống kê từ những thách thức đối với công nghệ phần mềm thì lỗi nhiều nhất là do

A/. Kiểm tra và bảo trì

B/. Phân tích yêu cầu (đ)

C/. Thiết kế

D/. Viết Code

2/. Yêu cầu có thể chia ra thành các loại nào sau đây?

A/. Chức năng, phi chức năng, yêu cầu hệ thống.

B/. Chức năng, phi chức năng (đ)

C/. Chức năng, phi chức năng, yêu cầu miền ứng dụng.

D/. Chức năng, phi chức năng, yêu cầu nghiệp vụ.

3/. 2 hình thức dùng mô tả yêu cầu là:

A/. Yêu cầu người dùng và yêu cầu hệ thống. (đ)

B/. Yêu cầu chức năng và yêu cầu phi chức năng.

C/. Yêu cầu chủ động và yêu cầu thụ động.

D/. Yêu cầu cụ thể và yêu cầu trừu tượng.

4/. Loại khả thi nào không được xem xét trong phân tích khả thi

A/. Khả thi về kinh tế.

B/. Khả thi về thực hiện.

C/. Khả thi về kỹ thuật.

D/. Khả thi về chất lượng . (đ)

5/. Tính chất cần có của dữ liệu trong phân tích yêu cầu

A/. Có định hướng thời gian. (đ)

B/. Có giá trị pháp lý.

C/. Tính mô tả trừu tượng

D/. Có thể mô tả bằng toán học.

6/. Câu hỏi nào có liên quan đến phân tích thiết kế?

A/. Thời gian hoàn thành dự án có đủ không?

B/. Làm thế nào chuyển thiết kế dữ liệu logic sang thiết kế dữ liệu vật lý? (đ)

C/. Các xử lý nào được tiến hành và các thông tin chi tiết liên quan?

D/. Đây là phạm vi của hệ thống phần mềm?

7/. Tính chất nào không cần thiết cho phân tích dữ liệu ?

A/. Cấu trúc dữ liệu.

B/. Đầy đủ.

C/. Bảo mật. (đ)

D/. Độ lớn.

8/. Phân tích yêu cầu bao gồm 3 hoạt động theo đúng thứ tự ?

A/. Làm tài liệu yêu cầu, làm rõ yêu cầu, xem xét yêu cầu.

- B/. Làm rõ yêu cầu, xem xét yêu cầu, làm tài liệu yêu cầu.**
C/. Xem xét yêu cầu, làm tài liệu yêu cầu, làm rõ yêu cầu.
D/. Làm rõ yêu cầu, làm tài liệu yêu cầu, xem xét yêu cầu.
- 9/. Làm rõ yêu cầu (*Eliciting requirements*) là
A/. Giao tiếp với khách hàng và người sử dụng để xác định các yêu cầu của họ.
B/. Các yêu cầu được ghi nhận lại theo nhiều hình thức.
C/. Các yêu cầu được tổng hợp lại theo nhiều hình thức.
D/. Xem các yêu cầu có ở tình trạng không rõ ràng?
- 10/. Yêu cầu nào là yêu cầu chức năng?
A/. Cảnh báo người dùng khi dung lượng trống trên đĩa còn 20%.
B/. Thực hiện thao tác thêm, xem, xóa, sửa dữ liệu nghiệp vụ.
C/. Cảnh báo ngày hệ thống bị sai.
D/. Yêu cầu chỉnh lại ngày giờ hệ thống mỗi khi làm việc.
- 11/. SRS là viết tắt của:
A/. Software Requirement Specification.
B/. System Requirement Specification.
C/. Studying Requirement Specification.
D/. Solve Requirement Specification.
- 12/. Phát biểu nào sau đây là không đúng khi nói đến quá trình thu thập yêu cầu:
A/. Yêu cầu rất khó phát hiện.
B/. Yêu cầu rất dễ bị thay đổi.
C/. Yêu cầu phải luôn thống nhất.
D/. Yêu cầu luôn được biết một cách chính xác. (đ)
- 13/. Kết quả của giai đoạn thu thập yêu cầu là:
A/. Bảng ước tính chi phí dự án
B/. Tài liệu đặc tả yêu cầu phần mềm. (đ)
C/. Lược đồ ngữ cảnh
D/. Lược đồ Use case và các lược đồ khác.
- 14/. Ai là người viết tài liệu SRS?
A/. Người quản lý dự án.
B/. Phân tích viên. (đ)
C/. Lập trình viên.
D/. Khách hàng.
- 15/. Kết quả cuối cùng của giai đoạn xác định và phân tích yêu cầu là:
A/. Tài liệu SRS. (đ)
B/. Sơ đồ DFD.
C/. Sơ đồ Use case
D/. Sơ đồ ERD.

- 16/. Mục nào sau đây không bao gồm trong tài liệu SRS?
- A/. Yêu cầu chức năng
 - B/. Yêu cầu phi chức năng
 - C/. Mục tiêu thực hiện
 - D/. Hướng dẫn sử dụng (đ)**
- 17/. Loại hình đặc tả nào được dùng phổ biến trong tài liệu SRS?
- A/. Đặc tả cấu trúc dữ liệu.
 - B/. Đặc tả chức năng.
 - C/. Đặc tả bằng sơ đồ. (đ)**
 - D/. Đặc tả đối tượng.
- 18/. Độ lớn (Volume) trong phân tích yêu cầu là:
- A/. Là số lượng máy tính chạy phần mềm.
 - B/. Là số lượng dữ liệu phát sinh trong một chu kỳ nào đó.
 - C/. Là số lượng các nghiệp vụ hệ thống phải tiến hành trong một chu kỳ nào đó. (đ)**
 - D/. Là số lượng người làm việc với phần mềm.
- 19/. Sơ đồ nào sau đây không cần thiết trong phân tích yêu cầu?
- A/. Use Case.
 - B/. Entity Relationship Diagram.
 - C/. State Transition Diagram.
 - D/. Activity Diagram. (đ)**
- 20/. Có bao nhiêu đặc trưng khi xem xét phân tích yêu cầu khả thi?
- A/. 2
 - B/. 3**
 - C/. 4
 - D/. 5
- 21/. Có bao nhiêu giai đoạn trong phân tích yêu cầu?
- A/. 3
 - B/. 4
 - C/. 5**
 - D/. 6
- 22/. Có bao nhiêu nguyên lý đặc tả yêu cầu?
- A/. 3
 - B/. 5
 - C/. 7
 - D/. 8**
- 23/. CASE là từ viết tắt của
- A/. Cost Aided Software Engineering.
 - B/. Computer Aided Software Engineering.**

- C/. Control Aided Software Engineering
D/. Computer Analyzing Software Engineering.
- 24/. Kỹ thuật thu thập yêu cầu nào cần đến chuyên gia?
A/. Interview.
B/. Observation.
C/. Expert
D/. Delphi.
- 25/. Kỹ thuật thu thập yêu cầu nào cần đến sự nhất trí của số đông?
A/. Prototype.
B/. Facilitated Workshops
C/. Observation
D/. Questionnaires & Surveys
- 26/. Mục nào không dùng cho đặc tả yêu cầu:
A/. Đặc tả cú pháp.
B/. Đặc tả đối tượng.
C/. Đặc tả chức năng.
D/. Đặc tả kỹ thuật.
- 27/. Mục nào không dùng cho đặc tả yêu cầu:
A/. Đặc tả thao tác.
B/. Đặc tả mô hình.
C/. Đặc tả bằng sơ đồ.
D/. Đặc tả thuật toán.
- 28/. Loại hình đặc tả nào không có?
A/. Đặc tả hình thức.
B/. Đặc tả phi hình thức.
C/. Đặc tả toán học.
D/. Đặc tả hỗn hợp.
- 29/. Xác nhận yêu cầu (Requirements Validation) được tiến hành bởi
A/. Phân tích viên và lập trình viên.
B/. Phân tích viên và khách hàng.
C/. Phân tích viên và các bên có liên quan.
D/. Phân tích viên và người dùng.
- 30/. Khi xác nhận yêu cầu, cần phải làm sáng tỏ các từ nào sau đây:
A/. “một số”, “đôi khi”, “thường”, “thông thường”, “bình thường”, “phần lớn”, “đa số”.
B/. Danh từ là số nhiều hay số ít.
C/. Tính từ chỉ trạng thái.
D/. Động từ ở hình thức chủ động hay bị động.
1. Câu hỏi không được kỹ sư phần mềm hiện nay quan tâm nữa

- a. **Tại sao chi phí phần cứng máy tính quá cao?**
 - b. Tại sao phần mềm mất một thời gian dài để hoàn tất?
 - c. Tại sao người ta tốn nhiều chi phí để phát triển một mẫu phần mềm?
 - d. Tại sao những lỗi phần mềm không được loại bỏ trong sản phẩm trước khi xuất xưởng
- 2. Mô hình phát triển ứng dụng nhanh
 - a. **definition, development, support**
 - b. what, how, where
 - c. programming, debugging, maintenance
 - d. analysis, design, testing
- 3. Mô hình phát triển ứng dụng nhanh
 - a. Một cách gọi khác của mô hình phát triển dựa vào thành phần
 - b. Một cách hữu dụng khi khách hàng không xác định yêu cầu rõ ràng
 - c. **Sự ráp nối tốc độ cao của mô hình tuần tự tuyến tính**
 - d. Tất cả mục trên
- 4. Mô hình tiến trình phần mềm tiến hóa
 - a. Bản chất lặp
 - b. Dễ dàng điều tiết những biến đổi yêu cầu sản phẩm
 - c. Nói chung không tạo ra những sản phẩm bỏ đi
 - d. **Tất cả các mục**
- 5. Mô hình phát triển phần mềm lặp lại tăng thêm
 - a. Một hướng hợp lý khi yêu cầu được xác định rõ
 - b. **Một hướng tốt khi cần tạo nhanh một sản phẩm thực thi lỗi**
 - c. Một hướng tốt nhất dùng cho những dự án có những nhóm phát triển lớn
 - d. Một mô hình cách mạng không nhưng không được dùng cho sản phẩm thương mại
- 6. Mô hình phát triển phần mềm xoắn ốc
 - a. Kết thúc với việc xuất xưởng sản phẩm phần mềm
 - b. Nhiều hỗn độn hơn với mô hình gia tăng
 - c. **Bao gồm việc đánh giá những rủi ro phần mềm trong mỗi vòng lặp**
 - d. Tất cả điều trên
- 7. Mô hình phát triển dựa vào thành phần
 - a. Chỉ phù hợp cho thiết kế phần cứng máy tính
 - b. Không thể hỗ trợ phát triển những thành phần sử dụng lại
 - c. **Dựa vào những kỹ thuật hỗ trợ đối tượng**
 - d. Không định chi phí hiệu quả bằng những độ đo phần mềm có thể định lượng
- 8. Để xây dựng mô hình hệ thống, kỹ sư phải quan tâm tới một trong những nhân tố hạn chế sau :
 - a. **Những giả định và những ràng buộc**
 - b. Ngân sách và phí tổn
 - c. Những đối tượng và những hoạt động
 - d. Lịch biểu và các mốc sự kiện
- 9. Trong kỹ thuật tiến trình nghiệp vụ, ba kiến trúc khác nhau được kiểm tra
 - a. **Hạ tầng kỹ thuật, dữ liệu, ứng dụng**
 - b. Hạ tầng tài chính, tổ chức và truyền thông
 - c. Cấu trúc báo cáo, cơ sở dữ liệu, mạng
 - d. Cấu trúc dữ liệu, yêu cầu, hệ thống
- 10. Thành phần nào của kỹ thuật tiến trình nghiệp vụ là trách nhiệm của kỹ sư phần mềm
 - a. Phân tích phạm vi nghiệp vụ

- b. **Thiết kế hệ thống nghiệp vụ**
 - c. Kế hoạch sản phẩm
 - d. Kế hoạch chiến lược thông tin
- 11. Những thành phần kiến trúc trong kỹ thuật sản phẩm là
 - a. **Dữ liệu, phần cứng, phần mềm, con người**
 - b. Dữ liệu, tài liệu, phần cứng, phần mềm
 - c. Dữ liệu, phần cứng, phần mềm, thủ tục
 - d. Tài liệu, phần cứng, con người, thủ tục
- 12. Đặc tả hệ thống mô tả
 - a. **Chức năng và hành vi của hệ thống dựa vào máy tính**
 - b. Việc thi hành của mỗi thành phần hệ thống được chỉ
 - c. Chi tiết giải thuật và cấu trúc hệ thống
 - d. Thời gian đòi hỏi cho việc giải lập hệ thống
- 13. Cách tốt nhất để đưa tới việc xem xét việc đánh giá yêu cầu là
 - a. Kiểm tra lỗi mô hình hệ thống
 - b. Nhờ khách hàng kiểm tra yêu cầu
 - c. Gởi họ tới đội thiết kế và xem họ có sự quan tâm nào không
 - d. **Dùng danh sách các câu hỏi kiểm tra để kiểm tra mỗi yêu cầu**
- 14. Sử dụng bảng lần vết giúp
 - a. Debug chương trình dựa theo việc phát hiện lỗi thời gian thực
 - b. Xác định việc biểu diễn những sự thi hành giải thuật
 - c. **Xác định, điều khiển và theo vết những thay đổi yêu cầu**
 - d. Không có mục nào
- 15. Mẫu mô hình hệ thống chứa thành phần
 - a. Input
 - b. Output
 - c. Giao diện người dùng
 - d. **Tất cả mục trên**
- 16. Tác vụ nào không được biểu diễn như là một phần của phân tích yêu cầu phần mềm
 - a. Định giá và tổng hợp
 - b. Mô hình hóa và thừa nhận vấn đề
 - c. **Lập kế hoạch và lịch biểu**
 - d. Đặc tả và xem xét
- 17. Đích của kỹ thuật đặc tả ứng dụng thuận tiện (FAST - facilitated application specification techniques) là nhờ người phát triển và khách hàng
 - a. Xây dựng một nguyên mẫu nhanh chóng
 - b. Học công việc lẫn nhau
 - c. **Làm việc với nhau để phát triển một tập những yêu cầu ban đầu**
 - d. Làm việc với nhau để phát triển những đặc tả phần mềm kỹ thuật
- 18. Ai là người không thích hợp để tham dự vào nhóm FAST (facilitated application specification techniques)
 - a. Kỹ sư phần cứng và phần mềm
 - b. Đại diện nhà sản xuất
 - c. Đại diện thị trường
 - d. **Nhân viên tài chánh cao cấp**
- 19. Những yêu cầu nào được quan tâm suốt QFD (quality function deployment)
 - a. exciting requirements
 - b. expected requirement
 - c. normal requirements
 - d. **technology requirements**

20. Phân tích giá trị được dẫn ra như là một phần của QFD (quality function deployment) nhằm xác định
- a. Chi phí của hoạt động đảm bảo chất lượng của dự án
 - b. Chi phí quan hệ của những yêu cầu qua việc triển khai chức năng, tác vụ và thông tin
 - c. **Độ ưu tiên quan hệ của những yêu cầu qua việc triển khai chức năng, tác vụ và thông tin**
 - d. Kích thước của bản ý kiến khách hàng
21. Use-cases là một kịch bản mà mô tả
- a. **Phần mềm thực hiện như thế nào khi được dùng trong một tình huống cho trước**
 - b. Những công cụ CASE sẽ được dùng như thế nào để xây dựng hệ thống
 - c. Kế hoạch xây dựng cho sản phẩm phần mềm
 - d. Những test-case cho sản phẩm phần mềm
22. Nội dung thông tin biểu diễn những đối tượng điều khiển và dữ liệu riêng biệt mà bao gồm những thông tin mà
- a. Cần thiết để trình bày tất cả output
 - b. Được đòi hỏi cho việc xử lý lỗi
 - c. Được đòi hỏi cho hoạt động tạo giao diện hệ thống
 - d. **Được biến đổi bởi phần mềm**
23. Dòng thông tin biểu diễn cách thức mà dữ liệu và điều khiển
- a. Quan hệ với một dữ liệu và điều khiển khác
 - b. **Biến đổi khi mỗi lần dịch chuyển qua hệ thống**
 - c. Sẽ được thực thi trong thiết kế cuối cùng
 - d. Không có mục nào
24. Cấu trúc thông tin biểu diễn tổ chức nội của
- a. Những cấu trúc dữ liệu dùng để biểu diễn loại dữ liệu
 - b. Mô hình bố trí nhân viên dự án
 - c. Mô hình truyền thông dự án
 - d. **Những dữ liệu khác nhau và những mục điều khiển**
25. Loại mô hình nào được tạo ra trong phân tích yêu cầu phần mềm
- a. **Chức năng và hành vi**
 - b. Giải thuật và cấu trúc dữ liệu
 - c. Kiến trúc và cấu trúc
 - d. Tính tin cậy và tính sử dụng
26. Trong ngữ cảnh của phân tích yêu cầu, hai loại phân tách vấn đề là
- a. bottom-up và top-down
 - b. **horizontal and vertical**
 - c. subordinate và superordinate
 - d. Không có mục nào
27. Khung nhìn (view) nào được quan tâm đầu tiên trong phân tích yêu cầu phần mềm
- a. actor view
 - b. data view
 - c. **essential view**
 - d. implementation view
28. Tạo nguyên mẫu tiến hóa thường thích được dùng hơn tạo nguyên mẫu bỏ đi bởi vì
- a. **Cho phép tái sử dụng nguyên mẫu đầu**
 - b. Không đòi hỏi làm việc nhiều với khách hàng
 - c. Dễ dàng thực hiện nhanh

- d. Nhiều tin cậy hơn
- 29. Những mục nào không là nguyên tắc cho việc biểu diễn yêu cầu
 - a. Biểu đồ phải thu hẹp về số và toàn vẹn trong sử dụng
 - b. Hình thức và nội dung biểu diễn thích hợp với nội dung
 - c. Những biểu diễn phải có thể xem xét lại
 - d. Dùng không hơn 7 màu dương và 2 màu âm trong biểu đồ**
- 30. Mục nào không là một mục đích cho việc xây dựng một mô hình phân tích
 - a. Xác định một tập những yêu cầu phần mềm
 - b. Mô tả yêu cầu khách hàng
 - c. Phát triển một giải pháp tóm tắt cho vấn đề**
 - d. Thiết lập một nền tảng cho thiết kế phần mềm
- 31. Sơ đồ luồng dữ liệu
 - a. Đưa ra hình ảnh quan hệ giữa các đối tượng dữ liệu
 - b. Đưa ra hình ảnh những chức năng biến đổi luồng dữ liệu**
 - c. Chỉ ra những quyết định logic chính khi chúng xuất hiện
 - d. Chỉ ra sự tương tác của hệ thống với sự kiện bên ngoài
- 32. Biểu đồ quan hệ thực thể
 - a. Đưa ra hình ảnh quan hệ giữa các đối tượng dữ liệu**
 - b. Đưa ra hình ảnh những chức năng biến đổi luồng dữ liệu
 - c. Chỉ ra những quyết định logic chính khi chúng xuất hiện
 - d. Chỉ ra sự tương tác của hệ thống với sự kiện bên ngoài
- 33. Biểu đồ dịch chuyển trạng thái
 - a. Đưa ra hình ảnh về các đối tượng dữ liệu
 - b. Đưa ra hình ảnh chức năng biến đổi luồng dữ liệu
 - c. Chỉ ra hình ảnh dữ liệu được biến đổi như thế nào bởi hệ thống
 - d. Chỉ ra những tương tác của hệ thống đối với sự kiện bên ngoài**
- 34. Phân tích văn phạm của bản tường thuật xử lý là bước đầu tiên tốt nhất để tạo ra
 - a. Tự điển dữ liệu
 - b. Biểu đồ dòng dữ liệu**
 - c. Biểu đồ quan hệ thực thể
 - d. Biểu đồ dịch chuyển trạng thái
- 35. Biểu đồ dòng điều khiển
 - a. Cần thiết để mô hình những hệ thống hướng sự kiện**
 - b. Được đòi hỏi cho tất cả hệ thống
 - c. Được dùng trong biểu đồ dòng dữ liệu
 - d. Hữu dụng trong mô hình hóa giao diện người dùng
- 36. Từ điển dữ liệu chứa những mô tả của mỗi
 - a. Mục cấu hình phần mềm
 - b. Đối tượng dữ liệu phần mềm**
 - c. Biểu đồ phần mềm
 - d. Hệ thống ký hiệu phần mềm
- 37. Mô hình thiết kế không quan tâm tới
 - a. Kiến trúc
 - b. Dữ liệu
 - c. Giao diện
 - d. Phạm vi dự án**
- 38. Sự quan trọng của thiết kế phần mềm có thể được tóm tắt bằng từ đơn
 - a. Accuracy
 - b. Complexity
 - c. Efficiency

- d. Quality**
39. Một đặc trưng của thiết kế tốt là
- a. Cho thấy sự liên kết mạnh giữa các module
 - b. Thực hiện tất cả yêu cầu trong phân tích**
 - c. Bao gồm những test case cho tất cả thành phần
 - d. Kết hợp mã nguồn nhằm mục đích mô tả
40. Mục nào không là đặc trưng chung trong các phương pháp thiết kế
- a. Quản lý cấu hình**
 - b. Ký hiệu thành phần chức năng
 - c. Nguyên tắc đánh giá chất lượng
 - d. Heuristic tinh chế
41. Loại trừu tượng nào được dùng trong thiết kế phần mềm
- a. Điều khiển
 - b. Dữ liệu
 - c. Thủ tục
 - d. Tất cả mục trên**
42. Loại mô hình nào không được có trong kiến trúc phần mềm
- a. Dữ liệu**
 - b. Động
 - c. Xử lý
 - d. Cấu trúc
43. Cấp bậc điều khiển thể hiện
- a. Thứ tự quyết định
 - b. Việc tổ chức của các module**
 - c. Sự lặp lại của những hoạt động
 - d. Sự tuần tự của các tiến trình
44. Thủ tục phần mềm tập trung vào
- a. Cấp bậc điều khiển trong một cảm nhận trừu tượng hơn
 - b. Xử lý chi tiết của mỗi module riêng biệt**
 - c. Xử lý chi tiết của mỗi tập module
 - d. Quan hệ giữa điều khiển và thủ tục
45. Nguyên nhân của việc sinh lỗi do thiết kế mức thành phần trước khi thiết kế dữ liệu là
- a. Thiết kế thành phần thì phụ thuộc vào ngôn ngữ còn thiết kế dữ liệu thì không
 - b. Thiết kế dữ liệu thì dễ thực hiện hơn
 - c. Thiết kế dữ liệu thì khó thực hiện
 - d. Cấu trúc dữ liệu thường ảnh hưởng tới cách thức mà thiết kế thành phần phải theo**
46. Mục đích của tham chiếu chéo những yêu cầu (ma trận) trong tài liệu thiết kế là nhằm
- a. Cho phép người quản lý theo dõi năng suất của nhóm thiết kế
 - b. Xác minh là tất cả các yêu cầu đã được xem xét trong thiết kế**
 - c. Chỉ ra chi phí kết hợp với mỗi yêu cầu
 - d. Cung cấp cho việc thực thi tên của những nhà thiết kế cho mỗi yêu cầu
47. Mục nào không là một phần của kiến trúc phần mềm
- a. Chi tiết giải thuật**
 - b. Cơ sở dữ liệu
 - c. Thiết kế dữ liệu
 - d. Cấu trúc chương trình
48. Đặc trưng nào là đúng cho kho dữ liệu, không phải là cơ sở dữ liệu đặc trưng

- a. Hướng mức nghiệp vụ và kích thước lớn
 - b. Thông tin đúng và hợp thời
 - c. Tích hợp và không thường thay đổi**
 - d. Tất cả những mục trên
49. Mẫu kiến trúc nhấn mạnh tới những thành phần
- a. Ràng buộc
 - b. Tập hợp những thành phần
 - c. Mô hình ngữ nghĩa
 - d. Tất cả những mục**
50. Nhằm xác định những mẫu kiến trúc hay kết hợp những mẫu phù hợp nhất cho hệ thống đề nghị, kỹ thuật yêu cầu dùng để khám phá
- a. Giải thuật phức tạp
 - b. Đặc trưng và ràng buộc**
 - c. Điều khiển và dữ liệu
 - d. Những mẫu thiết kế
51. Tiêu chuẩn đánh giá chất lượng của một thiết kế kiến trúc phải dựa vào
- a. Tính truy cập và tính tin cậy của hệ thống
 - b. Dữ liệu và điều khiển của hệ thống**
 - c. Tính chức năng của hệ thống
 - d. Những chi tiết thực thi của hệ thống
52. Trong phương pháp phân tích kiến trúc, mô tả mẫu kiến trúc thường dùng khung nhìn
- a. Dòng dữ liệu
 - b. Module
 - c. Tiến trình
 - d. Tất cả các mục trên**
53. Khi một luồng tổng thể trong một đoạn của biểu đồ luồng dữ liệu có tính trình tự cao và theo sau những những đường thẳng sẽ thể hiện
- a. Liên kết thấp
 - b. Module hóa tốt
 - c. Luồng giao dịch (transaction)
 - d. Luồng biến đổi (transform)**
54. Khi luồng thông tin trong một đoạn của sơ đồ luồng dữ liệu thể hiện bằng một mục đơn mà bấy một luồng dữ liệu khác theo một trong nhiều đường sẽ thể hiện
- a. Liên kết thấp
 - b. Module hóa tốt
 - c. Luồng giao dịch (transaction)**
 - d. Luồng biến đổi (transform)
55. Một bổ sung cần thiết nhằm biến đổi hay ánh xạ giao dịch để tạo một thiết kế kiến trúc đầy đủ là
- a. Sơ đồ quan hệ - thực thể
 - b. Từ điển dữ liệu
 - c. Mô tả việc xử lý cho mỗi module**
 - d. Những Test-case cho mỗi module
56. Những nguyên lý thiết kế giao diện nào không cho phép người dùng còn điều khiển tương tác với máy tính
- a. Cho phép được gián đoạn
 - b. Cho phép tương tác có thể undo
 - c. Che dấu những bản chất kỹ thuật với những người dùng thường
 - d. Chỉ cung cấp một cách thức xác định cứng khi hoàn thành tác vụ**
57. Những nguyên lý thiết kế giao diện cho phép người dùng ít phải nhớ

- a. Xác định những shortcut trực quan
 - b. Biểu lộ thông tin theo cách diễn tiến
 - c. Thiết lập những trường hợp mặc định có ý nghĩa
 - d. **Tất cả những mục trên**
58. Sự toàn vẹn (consistency) giao diện ngầm định
- a. Những kỹ thuật input giữ tương tự suốt ứng dụng
 - b. Mỗi ứng dụng phải có look and feel riêng biệt
 - c. Cách thức điều hướng (navigational) nhạy với ngữ cảnh
 - d. **Câu a và b**
59. Mô hình nào đưa ra hình ảnh tiền sử (profile) người dùng cuối của hệ thống dựa vào máy tính
- a. Mô hình thiết kế
 - b. **Mô hình người dùng**
 - c. Mô hình của người dùng
 - d. Mô hình nhận thức hệ thống
60. Mô hình nào đưa ra hình ảnh hệ thống trong đầu của người dùng cuối
- a. Mô hình thiết kế
 - b. Mô hình người dùng
 - c. Hình ảnh hệ thống
 - d. **Mô hình nhận thức hệ thống**
61. Mô hình nào đưa ra hình ảnh look and feel cho giao diện người dùng cùng những thông tin hỗ trợ
- a. Mô hình thiết kế
 - b. Mô hình người dùng
 - c. **Mô hình hình ảnh hệ thống**
 - d. Mô hình nhận thức hệ thống
62. Những hoạt động khung nào thường không kết hợp với những quá trình thiết kế giao diện người dùng
- a. **Ước lượng giá**
 - b. Xây dựng giao diện
 - c. Định trị giao diện
 - d. Phân tích người dùng và tác vụ
63. Hướng tiếp cận nào để những phân tích tác vụ của người dùng trong thiết kế giao diện người dùng
- a. Người dùng cho biết những ưa thích qua bản câu hỏi
 - b. Dựa vào ý kiến của những lập trình viên có kinh nghiệm
 - c. Nghiên cứu những hệ thống tự động liên quan
 - d. **Quan sát thao tác người dùng**
64. Những vấn đề thiết kế chung nổi trội lên trong hầu hết giao diện người dùng
- a. Kết nối tiền sử người dùng (profile) và shortcut chức năng
 - b. **Xử lý lỗi và thời gian đáp ứng của hệ thống**
 - c. Quyết định hiển thị hình ảnh và thiết kế icon
 - d. Không có mục nào
65. Những hệ thống phát triển giao diện người dùng đặc trưng cung cấp những kỹ thuật cho việc xây dựng những nguyên mẫu giao diện bao gồm
- a. Tạo code
 - b. Những tool vẽ
 - c. Định trị input
 - d. **Tất cả mục trên**
66. Những bản câu hỏi có ý nghĩa nhất đối với những người thiết kế giao diện khi được hoàn tất bởi

- a. Khách hàng
 - b. Những lập trình viên có kinh nghiệm
 - c. Người dùng sản phẩm**
 - d. Người quản lý dự án
67. Nhiều đo lường hữu dụng có thể thu thập khi quan sát những người dùng tương tác với hệ thống máy tính gồm
- a. Thời gian cho ứng dụng
 - b. Số khiếm khuyết (defect) phần mềm
 - c. Tính tin cậy của phần mềm
 - d. Thời gian đọc tài liệu trợ giúp**
68. Một bảng quyết định được dùng
- a. Để tư liệu tất cả những trạng thái phụ thuộc
 - b. Để hướng dẫn phát triển kế hoạch quản lý dự án
 - c. Chỉ khi xây dựng hệ chuyên gia
 - d. Khi một tập phức tạp những điều kiện và hoạt động xuất hiện trong thành phần**
69. Ngôn ngữ thiết kế chương trình (PDL) thường là một
- a. Sự kết hợp giữa cấu trúc lập trình và văn bản tường thuật**
 - b. Ngôn ngữ lập trình truyền thống theo luật riêng của nó
 - c. Ngôn ngữ phát triển phần mềm có thể đọc bởi máy
 - d. Một cách hữu dụng để biểu diễn kiến trúc phần mềm
70. Những độ đo phức tạp vòng (cyclomatic complexity metric) cung cấp cho người thiết kế thông tin về số
- a. Chu kỳ trong chương trình
 - b. Số lỗi trong chương trình
 - c. Những đường logic độc lập trong chương trình**
 - d. Những phát biểu của chương trình
71. Kiểm thử điều kiện là một kỹ thuật kiểm thử cấu trúc điều khiển mà những tiêu chuẩn dùng để thiết kế test-case
- a. Dựa vào kiểm thử đường cơ bản
 - b. Thử thách điều kiện logic trong module phần mềm**
 - c. Chọn những đường dẫn kiểm tra dựa vào những vị trí và dùng những biến
 - d. Tập trung vào việc kiểm thử việc giá trị những cấu trúc lặp
72. Kiểm thử luồng dữ liệu là một kỹ thuật kiểm thử cấu trúc điều khiển mà những tiêu chuẩn dùng để thiết kế test-case
- a. Dựa vào kiểm thử đường cơ bản
 - b. Thử thách điều kiện logic trong module phần mềm
 - c. Chọn những đường dẫn kiểm tra dựa vào những vị trí và dùng những biến**
 - d. Tập trung vào việc kiểm thử việc giá trị những cấu trúc lặp
73. Kiểm thử lặp là một kỹ thuật kiểm thử cấu trúc điều khiển mà những tiêu chuẩn dùng để thiết kế test-case
- a. Dựa vào kiểm thử đường cơ bản
 - b. Thử thách điều kiện logic trong module phần mềm
 - c. Chọn những đường dẫn kiểm tra dựa vào những vị trí và dùng những biến
 - d. Tập trung vào việc kiểm thử việc giá trị những cấu trúc lặp**
74. Kiểm thử Black-box cố gắng tìm ra những lỗi
- a. Chức năng không đầy đủ hay không đúng
 - b. Những lỗi giao diện

- c. Những lỗi thực thi
 - d. **Tất cả mục trên**
75. Lý do tốt nhất cho việc dùng nhóm kiểm tra phần mềm độc lập là
- a. Những người phát triển phần mềm không cần làm bất kỳ kiểm thử nào
 - b. Những người lạ sẽ kiểm phần mềm rất chặt
 - c. Những người kiểm thử không được đánh giá tới dự án cho đến khi kiểm thử bắt đầu
 - d. **Mâu thuẫn về quyền lợi giữa những người phát triển và những người kiểm thử sẽ giảm**
76. Trong một dự án thành công sử dụng chiến lược
- a. Đưa ra những xem xét kỹ thuật hình thức ưu tiên trước khi kiểm thử
 - b. Chỉ rõ những yêu cầu trong theo một cách thức có thể định lượng
 - c. Quan tâm tới việc sử dụng những nhóm kiểm thử độc lập
 - d. **Tất cả mục trên**
77. Kiểm thử tích hợp Top-down có thuận lợi chính là
- a. Những module mức thấp không bao giờ cần kiểm thử
 - b. **Những điểm quyết định chính được kiểm thử sớm**
 - c. Không có những stub cần phải viết
 - d. Không có mục nào
78. Kiểm thử tích hợp bottom-up có những thuận lợi chính
- a. Những điểm quyết định chính được kiểm thử sớm
 - b. Không có những driver cần được viết
 - c. **Không có những stub (nhánh) cần phải viết**
 - d. Không đòi hỏi kiểm thử hồi quy (regression)
79. Hướng debug
- a. Backtracking
 - b. Brute force
 - c. Sự loại trừ nguyên nhân
 - d. **Tất cả các mục**
80. Những kiểm tra chấp nhận thường được đưa ra bởi
- a. Người phát triển
 - b. **Những người dùng cuối**
 - c. Nhóm kiểm thử
 - d. Những kỹ sư hệ thống
81. Ai là người không thích hợp để tham dự vào nhóm FAST (facilitated application specification techniques)
- a. Kỹ sư phần cứng và phần mềm
 - b. Đại diện nhà sản xuất
 - c. Đại diện thị trường
 - d. **Nhân viên tài chính cao cấp**
82. Ba giai đoạn tổng quát của công nghệ phần mềm
- a. **definition, development, support**
 - b. what, how, where
 - c. programming, debugging, maintenance
 - d. analysis, design, testing
83. Biểu đồ dịch chuyển trạng thái
- a. Đưa ra hình ảnh về các đối tượng dữ liệu
 - b. Đưa ra hình ảnh chức năng biến đổi luồng dữ liệu
 - c. Chỉ ra hình ảnh dữ liệu được biến đổi như thế nào bởi hệ thống
 - d. **Chỉ ra những tương tác của hệ thống đối với sự kiện bên ngoài**

84. Biểu đồ dòng điều khiển
- Cần thiết để mô hình những hệ thống hướng sự kiện**
 - Được đòi hỏi cho tất cả hệ thống
 - Được dùng trong biểu đồ dòng dữ liệu
 - Hữu dụng trong mô hình hóa giao diện người dùng
85. Biểu đồ quan hệ thực thể
- Đưa ra hình ảnh quan hệ giữa các đối tượng dữ liệu**
 - Đưa ra hình ảnh những chức năng biến đổi luồng dữ liệu
 - Chỉ ra những quyết định logic chính khi chúng xuất hiện
 - Chỉ ra sự tương tác của hệ thống với sự kiện bên ngoài
86. Cách tốt nhất để đưa tới việc xem xét việc đánh giá yêu cầu là
- Kiểm tra lỗi mô hình hệ thống
 - Nhờ khách hàng kiểm tra yêu cầu
 - Gởi họ tới đội thiết kế và xem họ có sự quan tâm nào không
 - Dùng danh sách các câu hỏi kiểm tra để kiểm tra mỗi yêu cầu**
87. Cấp bậc điều khiển thể hiện
- Thứ tự quyết định
 - Việc tổ chức của các module**
 - Sự lặp lại của những hoạt động
 - Sự tuần tự của các tiến trình
88. Câu hỏi không được kỹ sư phần mềm hiện nay quan tâm nữa
- Tại sao chi phí phần cứng máy tính quá cao?**
 - Tại sao phần mềm mất một thời gian dài để hoàn tất?
 - Tại sao người ta tốn nhiều chi phí để phát triển một mẫu phần mềm?
 - Tại sao những lỗi phần mềm không được loại bỏ trong sản phẩm trước khi xuất xưởng
89. Cấu trúc thông tin biểu diễn tổ chức nội của
- Những cấu trúc dữ liệu dùng để biểu diễn loại dữ liệu
 - Mô hình bố trí nhân viên dự án
 - Mô hình truyền thông dự án
 - Những dữ liệu khác nhau và những mục điều khiển**
90. Chất lượng sản phẩm liên quan: product operation, product transition, product revision. Thuộc tính nào liên quan tới product revision:
- Reliability
 - Maintainability
 - Testability**
 - Portability
91. Chỉ phát biểu sai, bộ 3 ràng buộc
- Phạm vi
 - Thời gian
 - Chi phí
 - Chất lượng**
92. Chỉ phát biểu sai, các nhóm phần mềm (SUB-Team):
- Gồm một nhóm người
 - Sub-Team System analysis có nhiệm vụ ước tính lợi nhuận

- c. **Gồm một số người và nó phải tồn tại trong suốt dự án**
 - d. Có thể 1 người
93. Chỉ phát biểu sai, để đạt được độ đo PUM thấp:
- a. Cải tiến quy trình
 - b. Giảm lỗi giá
 - c. Gia tăng số bản bán được
 - d. **Giảm thời gian sửa lỗi**
94. Chỉ phát biểu sai. Kiểm thử áp lực
- a. Thường áp dụng trong hệ thống phân bố
 - b. **Nếu quá tải thiết kế thì không cần xem xét tới lỗi hệ thống**
 - c. Có thể xem là một dạng của kiểm thử thực thi
 - d. Thử thách dựa vào tải thiết kế cực đại
95. Chỉ phát biểu sai, lãnh vực hỗ trợ trong quản lý dự án
- a. Quản lý rủi ro
 - b. Quản lý mua sắm
 - c. **Quản lý tích hợp**
 - d. Quản lý truyền thông
96. Chỉ phát biểu sai. Mô hình hướng ngắt
- a. Cho phép đáp ứng nhanh
 - b. **Để lập trình**
 - c. Ít gây ra xung đột
 - d. Thường dùng trong hệ thống thời gian thực
97. Chỉ phát biểu sai. Phương pháp Brute Force
- a. “Để máy tính tìm ra lỗi”
 - b. Gần giống với phương pháp “vét cạn”
 - c. **Là một phương pháp hiệu quả**
 - d. Thường lặp đi lặp lại thủ tục đơn giản nhiều lần
98. Chỉ phát biểu sai. Thiết kế dữ liệu ở mức thành phần:
- a. **Thiết kế cơ sở dữ liệu**
 - b. Hiện thực thuộc tính dữ liệu thành cấu trúc dữ liệu
 - c. Phát triển một tập những trừu tượng dữ liệu
 - d. Tinh chế các đối tượng dữ liệu
99. Chỉ phát biểu sai. V & V (Verification and Validation)
- a. Đánh giá hệ thống có tính sử dụng hay không
 - b. Liên quan tới vấn đề debug và bảo mật
 - c. **Nó và kiểm thử là hai lãnh vực riêng**
 - d. Nhằm kiểm tra phần mềm phải thực hiện những gì người dùng thực sự cần
100. Chỉ ra mục sai. Trong mô hình WebE trong mô hình phân tích có
- a. Phân tích nội dung
 - b. Phân tích cấu hình
 - c. Phân tích tương tác
 - d. **Phân tích điều hướng (naviation)**
101. Chỉ ra phát biểu sai. Help:
- a. Có nhiều điểm vào nên người dùng có thể vào hệ thống help từ nhiều nơi
 - b. Help! Nghĩa là “Help. I’m in trouble”
 - c. **Help như một số tay hướng dẫn on-line**

- d. Những chỉ báo cho biết vị trí của người dùng trong hệ thống help
102. Chỉ ra phát biểu sai. Quá trình kiểm nghiệm phần mềm
- Phải có khả năng tìm ra lỗi cao
 - Phải có tính chọn lọc
 - Nhằm xác định phần mềm không có lỗi**
 - Không nên dư thừa và quá phức tạp
103. Chọn 5 hoạt động chính, tổng quát trong quá trình xây dựng phần mềm
- Giao tiếp, lập kế hoạch, mô hình hóa, xây dựng, triển khai**
 - Phân tích, thiết kế, lập trình, gỡ lỗi, bảo trì
 - Không có mục nào
 - Giao tiếp, quản lý rủi ro, ước lượng, sản xuất, kiểm tra lại
104. Có mấy loại vòng lặp
- 4**
 - 3
 - 2
 - 5
105. Công nghệ Web có những đặc điểm
- Nó thường dùng mô hình gia tăng (incremental process model)
 - Thời gian chuyển giao sản phẩm rất nhanh
 - Những thay đổi (change) diễn ra nhanh chóng
 - Nó là một công nghệ mới, nó cần phải tách xa công nghệ trước đây**
106. Dòng thông tin biểu diễn cách thức mà dữ liệu và điều khiển
- Quan hệ với một dữ liệu và điều khiển khác
 - Biến đổi khi mỗi lần dịch chuyển qua hệ thống**
 - Sẽ được thực thi trong thiết kế cuối cùng
 - Không có mục nào
107. Đặc điểm nào sau đây được sử dụng để đánh giá một bản thiết kế tốt?
- Thể hiện tất cả các yêu cầu trong pha phân tích
 - Chứa cả các trường hợp kiểm thử của tất cả các thành phần
 - Cung cấp một mô tả hoàn thiện về phần mềm
 - Câu a và c**
108. Đặc tả hệ thống mô tả
- Chức năng và hành vi của hệ thống dựa vào máy tính**
 - Việc thi hành của mỗi thành phần hệ thống được chỉ
 - Chi tiết giải thuật và cấu trúc hệ thống
 - Thời gian đòi hỏi cho việc giả lập hệ thống
109. Đặc trưng nào là đúng cho kho dữ liệu, không phải là cơ sở dữ liệu đặc trưng
- Hướng mức nghiệp vụ và kích thước lớn
 - Thông tin đúng và hợp thời
 - Tích hợp và không thường thay đổi**
 - Tất cả những mục trên
110. Để xây dựng mô hình hệ thống, kỹ sư phải quan tâm tới một trong những nhân tố hạn chế sau:
- Những giả định và những ràng buộc**
 - Ngân sách và phí tổn
 - Những đối tượng và những hoạt động

- d. Lịch biểu và các mốc sự kiện
111. Đích của kỹ thuật đặc tả ứng dụng thuận tiện (FAST - facilitated application specification techniques) là nhờ người phát triển và khách hàng
- Xây dựng một nguyên mẫu nhanh chóng
 - Học công việc lẫn nhau
 - Làm việc với nhau để phát triển một tập những yêu cầu ban đầu**
 - Làm việc với nhau để phát triển những đặc tả phần mềm kỹ thuật
112. Độ đo chất lượng của việc khử lỗi là tỷ số của
- Số lần giải quyết trong tháng và tổng số vấn đề phát sinh trong tháng
 - Số lần bảo trì vượt thời gian quy định và tổng số lần bảo trì
 - Tất cả đều đúng
 - Số lần sai lầm trong việc sửa lỗi và tổng số lần sửa lỗi**
113. FP (Function Point) là
- Độ đo sản phẩm cuối
 - Độ đo năng suất phần mềm hướng kích thước
 - Tất cả đều sai**
 - Độ đo năng suất phần mềm hướng chức năng
114. Hình thức kiểm nghiệm tích hợp hướng đối tượng
- Kiểm nghiệm trên cơ sở thread
 - Kiểm nghiệm trên cơ sở sử dụng
 - Câu a, b đúng**
 - Câu a, b sai
115. Hướng debug
- Backtracking
 - Brute force
 - Sự loại trừ nguyên nhân
 - Tất cả các mục**
116. Hướng tiếp cận nào để những phân tích tác vụ của người dùng trong thiết kế giao diện người dùng
- Người dùng cho biết những ưa thích qua bản câu hỏi
 - Dựa vào ý kiến của những lập trình viên có kinh nghiệm
 - Nghiên cứu những hệ thống tự động liên quan
 - Quan sát thao tác người dùng**
117. Khả năng được chấp nhận trong các yêu cầu đối với phần mềm
- Là tính tin cậy
 - Tất cả đều sai
 - Là sự chấp nhận được về giao diện
 - Là sự phù hợp với yêu cầu người sử dụng**
118. Khi luồng thông tin trong một đoạn của sơ đồ luồng dữ liệu thể hiện bằng một mục đơn mà bấy một luồng dữ liệu khác theo một trong nhiều đường sẽ thể hiện
- Liên kết thấp
 - Module hóa tốt
 - Luồng giao dịch (transaction)**
 - Luồng biến đổi (transform)
119. Khi một luồng tổng thể trong một đoạn của biểu đồ luồng dữ liệu có tính trình tự cao và theo sau những những đường thẳng sẽ thể hiện

- a. Liên kết thấp
 - b. Module hóa tốt
 - c. Luồng giao dịch (transaction)
 - d. **Luồng biến đổi (transform)**
120. Khung nhìn (view) nào được quan tâm đầu tiên trong phân tích yêu cầu phần mềm
- a. actor view
 - b. data view
 - c. **essential view**
 - d. implementation view
121. Kiểm nghiệm hướng đối tượng thường dùng
- a. **Kiểm nghiệm tích hợp đối tượng**
 - b. Kiểm nghiệm hộp đen
 - c. Kiểm nghiệm thừa kế
 - d. Kiểm nghiệm hộp trắng
122. Kiểm nghiệm tích hợp module
- a. **Có thể tích hợp từ trên xuống dưới**
 - b. Tất cả đều đúng
 - c. Có thể tích hợp từ dưới lên theo cách tích hợp theo chiều ngang
 - d. Có thể tích hợp từ dưới lên theo cách tích hợp theo chiều sâu
123. Kiểm thử Black-box cố gắng tìm ra những lỗi
- a. Chức năng không đầy đủ hay không đúng
 - b. Những lỗi giao diện
 - c. Những lỗi thực thi
 - d. **Tất cả mục trên**
124. Kiểm thử điều kiện là một kỹ thuật kiểm thử cấu trúc điều khiển mà những tiêu chuẩn dùng để thiết kế test-case
- a. Dựa vào kiểm thử đường cơ bản
 - b. **Thử thách điều kiện logic trong module phần mềm**
 - c. Chọn những đường dẫn kiểm tra dựa vào những vị trí và dùng những biến
 - d. Tập trung vào việc kiểm thử việc giá trị những cấu trúc lặp
125. Kiểm thử lặp là một kỹ thuật kiểm thử cấu trúc điều khiển mà những tiêu chuẩn dùng để thiết kế test-case
- a. Dựa vào kiểm thử đường cơ bản
 - b. Thử thách điều kiện logic trong module phần mềm
 - c. Chọn những đường dẫn kiểm tra dựa vào những vị trí và dùng những biến
 - d. **Tập trung vào việc kiểm thử việc giá trị những cấu trúc lặp**
126. Kiểm thử luồng dữ liệu là một kỹ thuật kiểm thử cấu trúc điều khiển mà những tiêu chuẩn dùng để thiết kế test-case
- a. Dựa vào kiểm thử đường cơ bản
 - b. Thử thách điều kiện logic trong module phần mềm
 - c. **Chọn những đường dẫn kiểm tra dựa vào những vị trí và dùng những biến**
 - d. Tập trung vào việc kiểm thử việc giá trị những cấu trúc lặp
127. Kiểm thử tích hợp bottom-up có những thuận lợi chính

- a. Những điểm quyết định chính được kiểm thử sớm
 - b. Không có những driver cần được viết
 - c. **Không có những stub (nhánh) cần phải viết**
 - d. Không đòi hỏi kiểm thử hồi quy (regression)
128. Kiểm thử tích hợp Top-down có thuận lợi chính là
- a. Những module mức thấp không bao giờ cần kiểm thử
 - b. **Những điểm quyết định chính được kiểm thử sớm**
 - c. Không có những stub cần phải viết
 - d. Không có mục nào
129. Kiểm thử vòng lặp lồng nhau
- a. Tất cả đều đúng
 - b. **Khi xét vòng lặp nào thì cần test Min+1, typical, max-1**
 - c. Kiểm thử từ ngoài vào trong
 - d. Nếu các vòng lặp là độc lập thì xem như là vòng lặp đơn
130. Liên kết (Coupling) là một chỉ báo chất lượng cho biết mức độ mà module
- a. Tập trung vào chỉ một điều
 - b. **Kết nối với module khác và môi trường bên ngoài**
 - c. Có thể hoàn thành chức năng của nó trong một cách thức phù hợp về thời gian
 - d. Có thể được viết với sự rần chắc nhiều hơn
131. Loại mô hình nào được tạo ra trong phân tích yêu cầu phần mềm
- a. **Chức năng và hành vi**
 - b. Giải thuật và cấu trúc dữ liệu
 - c. Kiến trúc và cấu trúc
 - d. Tính tin cậy và tính sử dụng
132. Loại mô hình nào không được có trong kiến trúc phần mềm
- a. **Dữ liệu**
 - b. Động
 - c. Xử lý
 - d. Cấu trúc
133. Loại trừu tượng nào được dùng trong thiết kế phần mềm
- a. Điều khiển
 - b. Dữ liệu
 - c. Thủ tục
 - d. **Tất cả mục trên**
134. Lý do tốt nhất cho việc dùng nhóm kiểm tra phần mềm độc lập là
- a. Những người phát triển phần mềm không cần làm bất kỳ kiểm thử nào
 - b. Những người lạ sẽ kiểm phần mềm rất chặt
 - c. Những người kiểm thử không được dính dáng tới dự án cho đến khi kiểm thử bắt đầu
 - d. **Mâu thuẫn về quyền lợi giữa những người phát triển và những người kiểm thử sẽ giảm**
135. Mật độ lỗi (defect density) thuộc độ đo
- a. **Độ đo chất lượng sản phẩm cuối**
 - b. Độ đo quá trình sản xuất
 - c. Độ đo dự án phần mềm
 - d. Độ đo chất lượng bảo trì

136. Mẫu kiến trúc nhấn mạnh tới những thành phần
- Ràng buộc
 - Tập hợp những thành phần
 - Mô hình ngữ nghĩa
 - Tất cả những mục**
137. Mẫu mô hình hệ thống chứa thành phần
- Input
 - Output
 - Giao diện người dùng
 - Tất cả mục trên**
138. Milestone
- Thường có output là những word product
 - Là thời điểm cuối của một hoạt động xử lý
 - Có thể chuyển giao một kết quả của dự án
 - Tất cả đều đúng**
139. Mô hình nào đưa ra hình ảnh hệ thống trong đầu của người dùng cuối
- Mô hình thiết kế
 - Mô hình người dùng
 - Hình ảnh hệ thống
 - Mô hình nhận thức hệ thống**
140. Mô hình nào đưa ra hình ảnh look and feel cho giao diện người dùng cùng những thông tin hỗ trợ
- Mô hình thiết kế
 - Mô hình người dùng
 - Mô hình hình ảnh hệ thống**
 - Mô hình nhận thức hệ thống
141. Mô hình nào đưa ra hình ảnh tiền sử (profile) người dùng cuối của hệ thống dựa vào máy tính
- Mô hình thiết kế
 - Mô hình người dùng**
 - Mô hình của người dùng
 - Mô hình nhận thức hệ thống
142. Mô hình phát triển dựa vào thành phần
- Chỉ phù hợp cho thiết kế phần cứng máy tính
 - Không thể hỗ trợ phát triển những thành phần sử dụng lại
 - Dựa vào những kỹ thuật hỗ trợ đối tượng**
 - Không định chi phí hiệu quả bằng những độ đo phần mềm có thể định lượng
143. Mô hình phát triển phần mềm dựa trên mẫu thử là
- Một mô hình rất rủi ro, khó đưa ra được một sản phẩm tốt
 - Phương pháp tốt nhất được sử dụng trong các dự án có nhiều thành viên
 - Một phương pháp hữu ích khi khách hàng không thể xác định yêu cầu một cách rõ ràng**
 - Một phương pháp thích hợp được sử dụng khi các yêu cầu đã được xác định rõ ràng
144. Mô hình phát triển phần mềm lặp lại tăng thêm
- Một hướng hợp lý khi yêu cầu được xác định rõ

- b. **Một hướng tốt khi cần tạo nhanh một sản phẩm thực thi lỗi**
 c. Một hướng tốt nhất dùng cho những dự án có những nhóm phát triển lớn
 d. Một mô hình cách mạng không nhưng không được dùng cho sản phẩm thương mại
145. Mô hình phát triển phần mềm tuần tự tuyến tính còn gọi là
 a. Mô hình hỗn độn
 b. Mô hình nước suối
 c. Mô hình xoắn ốc
 d. **Mô hình chu kỳ sống cổ điển**
146. Mô hình phát triển phần mềm xoắn ốc
 a. Kết thúc với việc xuất xưởng sản phẩm phần mềm
 b. Nhiều hỗn độn hơn với mô hình gia tăng
 c. **Bao gồm việc đánh giá những rủi ro phần mềm trong mỗi vòng lặp**
 d. Tất cả điều trên
147. Mô hình phát triển ứng dụng nhanh
 a. Một cách gọi khác của mô hình phát triển dựa vào thành phần
 b. Một cách hữu dụng khi khách hàng không xác định yêu cầu rõ ràng
 c. **Sự ráp nối tốc độ cao của mô hình tuần tự tuyến tính**
 d. Tất cả mục trên
148. Mô hình thiết kế không quan tâm tới
 a. Kiến trúc
 b. Dữ liệu
 c. Giao diện
 d. **Phạm vi dự án**
149. Mô hình tiến trình phần mềm tiến hóa
 a. Bản chất lặp
 b. Dễ dàng điều tiết những biến đổi yêu cầu sản phẩm
 c. Nói chung không tạo ra những sản phẩm bỏ đi
 d. **Tất cả các mục**
150. Một bảng quyết định được dùng
 a. Để tư liệu tất cả những trạng thái phụ thuộc
 b. Để hướng dẫn phát triển kế hoạch quản lý dự án
 c. Chỉ khi xây dựng hệ chuyên gia
 d. **Khi một tập phức tạp những điều kiện và hoạt động xuất hiện trong thành phần**
151. Một bổ sung cần thiết nhằm biến đổi hay ánh xạ giao dịch để tạo một thiết kế kiến trúc đầy đủ là
 a. Sơ đồ quan hệ - thực thể
 b. Từ điển dữ liệu
 c. **Mô tả việc xử lý cho mỗi module**
 d. Những Test-case cho mỗi module
152. Một đặc trưng của thiết kế tốt là
 a. Cho thấy sự liên kết mạnh giữa các module
 b. **Thực hiện tất cả yêu cầu trong phân tích**
 c. Bao gồm những test case cho tất cả thành phần
 d. Kết hợp mã nguồn nhằm mục đích mô tả

153. Mục đích của tham chiếu chéo những yêu cầu (ma trận) trong tài liệu thiết kế là nhằm
- Cho phép người quản lý theo dõi năng suất của nhóm thiết kế
 - Xác minh là tất cả các yêu cầu đã được xem xét trong thiết kế**
 - Chỉ ra chi phí kết hợp với mỗi yêu cầu
 - Cung cấp cho việc thực thi tên của những nhà thiết kế cho mỗi yêu cầu
154. Mục nào không là đặc trưng chung trong các phương pháp thiết kế
- Quản lý cấu hình**
 - Ký hiệu thành phần chức năng
 - Nguyên tắc đánh giá chất lượng
 - Heuristic tinh chế
155. Mục nào không là một mẫu kiến trúc (pattern)? Mẫu
- Concurrency
 - Persistence
 - Distribution
 - Borker**
156. Mục nào không là một mục đích cho việc xây dựng một mô hình phân tích
- Xác định một tập những yêu cầu phần mềm
 - Mô tả yêu cầu khách hàng
 - Phát triển một giải pháp tóm tắt cho vấn đề**
 - Thiết lập một nền tảng cho thiết kế phần mềm
157. Mục nào không là một phần của kiến trúc phần mềm
- Chi tiết giải thuật**
 - Cơ sở dữ liệu
 - Thiết kế dữ liệu
 - Cấu trúc chương trình
158. Mục nào không phải là một loại kiến trúc (style): kiến trúc
- Luồng dữ liệu
 - Kiến trúc ngữ cảnh**
 - Gọi trả về
 - Tầng
159. Mục nào liên quan tới phân tích người dùng:
- Mô hình hệ thống của người dùng
 - Trong tình huống đặc trưng thì người dùng thực hiện công việc gì?
 - Những feedback từ việc đánh giá của người dùng
 - Nếu người dùng xảy ra lỗi thì hậu quả như thế nào?**
160. Ngôn ngữ thiết kế chương trình (PDL) thường là một
- Sự kết hợp giữa cấu trúc lập trình và văn bản tường thuật**
 - Ngôn ngữ lập trình truyền thống theo luật riêng của nó
 - Ngôn ngữ phát triển phần mềm có thể đọc bởi máy
 - Một cách hữu dụng để biểu diễn kiến trúc phần mềm
161. Nguyên nhân của việc sinh lỗi do thiết kế mức thành phần trước khi thiết kế dữ liệu là
- Thiết kế thành phần thì phụ thuộc vào ngôn ngữ còn thiết kế dữ liệu thì không
 - Thiết kế dữ liệu thì dễ thực hiện hơn
 - Thiết kế dữ liệu thì khó thực hiện

d. Cấu trúc dữ liệu thường ảnh hưởng tới cách thức mà thiết kế thành phần phải theo

162. Nhằm xác định những mẫu kiến trúc hay kết hợp những mẫu phù hợp nhất cho hệ thống đề nghị, kỹ thuật yêu cầu dùng để khám phá

- a. Giải thuật phức tạp
- b. Đặc trưng và ràng buộc**
- c. Điều khiển và dữ liệu
- d. Những mẫu thiết kế

163. Nhiều đo lường hữu dụng có thể thu thập khi quan sát những người dùng tương tác với hệ thống máy tính gồm

- a. Thời gian cho ứng dụng
- b. Số khiếm khuyết (defect) phần mềm
- c. Tính tin cậy của phần mềm
- d. Thời gian đọc tài liệu trợ giúp**

164. Những bản câu hỏi có ý nghĩa nhất đối với những người thiết kế giao diện khi được hoàn tất bởi

- a. Khách hàng
- b. Những lập trình viên có kinh nghiệm
- c. Người dùng sản phẩm**
- d. Người quản lý dự án

165. Những độ đo phức tạp vòng (cyclomatic complexity metric) cung cấp cho người thiết kế thông tin về số

- a. Chu kỳ trong chương trình
- b. Số lỗi trong chương trình
- c. Những đường logic độc lập trong chương trình**
- d. Những phát biểu của chương trình

166. Những gì làm cho khó đưa ra những yêu cầu

- a. Hiểu rõ những yêu cầu người dùng
- b. Sự thay đổi
- c. Tất cả các mục**
- d. Phạm vi, giới hạn

167. Những hệ thống phát triển giao diện người dùng đặc trưng cung cấp những kỹ thuật cho việc xây dựng những nguyên mẫu giao diện bao gồm

- a. Tạo code
- b. Những tool vẽ
- c. Định trị input
- d. Tất cả các mục trên**

168. Những hoạt động khung nào thường không kết hợp với những quá trình thiết kế giao diện người dùng

- a. Ước lượng giá**
- b. Xây dựng giao diện
- c. Định trị giao diện
- d. Phân tích người dùng và tác vụ

169. Những kiểm tra chấp nhận thường được đưa ra bởi

- a. Người phát triển
- b. Những người dùng cuối**
- c. Nhóm kiểm thử

- d. Những kỹ sư hệ thống
170. Những mục nào không là nguyên tắc cho việc biểu diễn yêu cầu
- Biểu đồ phải thu hẹp về số và toàn vẹn trong sử dụng
 - Hình thức và nội dung biểu diễn thích hợp với nội dung
 - Những biểu diễn phải có thể xem xét lại
 - Dùng không hơn 7 màu dương và 2 màu âm trong biểu đồ**
171. Những nguyên lý thiết kế giao diện cho phép người dùng ít phải nhớ
- Xác định những shortcut trực quan
 - Biểu lộ thông tin theo cách diễn tiến
 - Thiết lập những trường hợp mặc định có ý nghĩa
 - Tất cả những mục trên**
172. Những nguyên lý thiết kế giao diện nào không cho phép người dùng còn điều khiển tương tác với máy tính
- Cho phép được gián đoạn
 - Cho phép tương tác có thể undo
 - Che dấu những bản chất kỹ thuật với những người dùng thường
 - Chỉ cung cấp một cách thức xác định cứng khi hoàn thành tác vụ**
173. Những thành phần kiến trúc trong kỹ thuật sản phẩm là
- Dữ liệu, phần cứng, phần mềm, con người**
 - Dữ liệu, tài liệu, phần cứng, phần mềm
 - Dữ liệu, phần cứng, phần mềm, thủ tục
 - Tài liệu, phần cứng, con người, thủ tục
174. Những yêu cầu nào được quan tâm suốt QFD (quality function deployment)
- exciting requirements
 - expected requirement
 - normal requirements
 - technology requirements**
175. Những vấn đề thiết kế chung nổi trội lên trong hầu hết giao diện người dùng
- Kết nối tiền sử người dùng (profile) và shortcut chức năng
 - Xử lý lỗi và thời gian đáp ứng của hệ thống**
 - Quyết định hiển thị hình ảnh và thiết kế icon
 - Không có mục nào
176. Nội dung thông tin biểu diễn những đối tượng điều khiển và dữ liệu riêng biệt mà bao gồm những thông tin mà
- Cần thiết để trình bày tất cả output
 - Được đòi hỏi cho việc xử lý lỗi
 - Được đòi hỏi cho hoạt động tạo giao diện hệ thống
 - Được biến đổi bởi phần mềm**
177. Phân tích giá trị được dẫn ra như là một phần của QFD (quality function deployment) nhằm xác định
- Chi phí của hoạt động đảm bảo chất lượng của dự án
 - Chi phí quan hệ của những yêu cầu qua việc triển khai chức năng, tác vụ và thông tin
 - Độ ưu tiên quan hệ của những yêu cầu qua việc triển khai chức năng, tác vụ và thông tin**
 - Kích thước của bản ý kiến khách hàng

178. Phân tích văn phạm của bản tường thuật xử lý là bước đầu tiên tốt nhất để tạo ra
- Tự diễn dữ liệu
 - Biểu đồ dòng dữ liệu**
 - Biểu đồ quan hệ thực thể
 - Biểu đồ dịch chuyển trạng thái
179. Sơ đồ luồng dữ liệu
- Đưa ra hình ảnh quan hệ giữa các đối tượng dữ liệu
 - Đưa ra hình ảnh những chức năng biến đổi luồng dữ liệu**
 - Chỉ ra những quyết định logic chính khi chúng xuất hiện
 - Chỉ ra sự tương tác của hệ thống với sự kiện bên ngoài
180. Sử dụng bảng lần vết giúp
- Debug chương trình dựa theo việc phát hiện lỗi thời gian thực
 - Xác định việc biểu diễn những sự thi hành giải thuật
 - Xác định, điều khiển và theo vết những thay đổi yêu cầu**
 - Không có mục nào
181. Sự quan trọng của thiết kế phần mềm có thể được tóm tắt bằng từ đơn
- Accuracy
 - Complexity
 - Efficiency
 - Quality**
182. Sự toàn vẹn (consistency) giao diện ngầm định
- Những kỹ thuật input giữ tương tự suốt ứng dụng
 - Mỗi ứng dụng phải có look and feel riêng biệt
 - Cách thức điều hướng (navigational) nhạy với ngữ cảnh
 - Câu a và b**
183. Tác vụ nào không được biểu diễn như là một phần của phân tích yêu cầu phần mềm
- Định giá và tổng hợp
 - Mô hình hóa và thừa nhận vấn đề
 - Lập kế hoạch và lịch biểu**
 - Đặc tả và xem xét
184. Tài liệu nào sau đây sẽ được tạo ra trong pha thiết kế hệ thống?
- Kế hoạch kiểm thử
 - Mã lệnh
 - Thiết kế chi tiết**
 - Lập kế hoạch
185. Tạo nguyên mẫu tiến hóa thường thích được dùng hơn tạo nguyên mẫu bỏ đi bởi vì
- Cho phép tái sử dụng nguyên mẫu đầu**
 - Không đòi hỏi làm việc nhiều với khách hàng
 - Dễ dàng thực hiện nhanh
 - Nhiều tin cậy hơn
186. Thành phần nào của kỹ thuật tiến trình nghiệp vụ là trách nhiệm của kỹ sư phần mềm
- Phân tích phạm vi nghiệp vụ
 - Thiết kế hệ thống nghiệp vụ**

- c. Kế hoạch sản phẩm
 - d. Kế hoạch chiến lược thông tin
187. Theo Boris Beizer, thiết kế Testcase cần theo ràng buộc (constraint)
- a. Theo một cách thức đầy đủ
 - b. Tất cả đều đúng**
 - c. Nỗ lực và thời gian là tối thiểu
 - d. Nhằm khám phá lỗi
188. Theo chiến thuật kiểm nghiệm phổ biến, kiểm nghiệm tính năng tương quan với
- a. Phân tích toàn bộ hệ thống
 - b. Thiết kế
 - c. Phân tích yêu cầu**
 - d. Mã hóa
189. Thủ tục phần mềm tập trung vào
- a. Cấp bậc điều khiển trong một cảm nhận trừu tượng hơn
 - b. Xử lý chi tiết của mỗi module riêng biệt**
 - c. Xử lý chi tiết của mỗi tập module
 - d. Quan hệ giữa điều khiển và thủ tục
190. Tiêu chuẩn đánh giá chất lượng của một thiết kế kiến trúc phải dựa vào
- a. Tính truy cập và tính tin cậy của hệ thống
 - b. Dữ liệu và điều khiển của hệ thống**
 - c. Tính chức năng của hệ thống
 - d. Những chi tiết thực thi của hệ thống
191. Tiêu chuẩn ISO để hướng dẫn thực hiện cho lĩnh vực phần mềm là
- a. ISO 9001
 - b. Tất cả đều sai
 - c. ISO 15288
 - d. ISO 9000-3**
192. Trong biểu diễn lịch biểu dự án Critical path là đường
- a. Là một đường duy nhất
 - b. Có thời gian ngắn nhất
 - c. Có thời gian dài nhất**
 - d. Tất cả đều đúng phụ thuộc vào dự án
193. Trong độ đo hiệu quả khử lỗi DRE, số lỗi tiềm tàng là
- a. Tất cả đều sai
 - b. Số lỗi do khách hàng phát hiện
 - c. Toàn bộ lỗi được phát hiện sau đó**
 - d. Toàn bộ lỗi chưa phát hiện
194. Trong kỹ thuật tiến trình nghiệp vụ, ba kiến trúc khác nhau được kiểm tra
- a. Hạ tầng kỹ thuật, dữ liệu, ứng dụng**
 - b. Hạ tầng tài chính, tổ chức và truyền thông
 - c. Cấu trúc báo cáo, cơ sở dữ liệu, mạng
 - d. Cấu trúc dữ liệu, yêu cầu, hệ thống
195. Trong mô hình CMM (Software Capability Maturity Model) có mấy mức độ trưởng thành
- a. 5 mức độ**
 - b. 4 mức độ

- c. 6 mức độ
 - d. 3 mức độ
196. Trong mô hình phân tích thành phần dựa vào kịch bản (Scenario based element) được dùng cho
- a. Thiết kế kiến trúc
 - b. Thiết kế thành phần
 - c. **Thiết kế giao diện**
 - d. Thiết kế dữ liệu/class
197. Trong một dự án thành công sử dụng chiến lược
- a. Đưa ra những xem xét kỹ thuật hình thức ưu tiên trước khi kiểm thử
 - b. Chỉ rõ những yêu cầu trong theo một cách thức có thể định lượng
 - c. Quan tâm tới việc sử dụng những nhóm kiểm thử độc lập
 - d. **Tất cả mục trên**
198. Trong ngữ cảnh của phân tích yêu cầu, hai loại phân tách vấn đề là
- a. bottom-up và top-down
 - b. **horizontal and vertical**
 - c. subordinate và superordinate
 - d. Không có mục nào
199. Trong nhận diện rủi ro, việc không đáp ứng về lịch biểu thuộc loại rủi ro
- a. Về con người
 - b. **Về ước lượng**
 - c. Về yêu cầu
 - d. Về tổ chức
200. Trong phương pháp phân tích kiến trúc, mô tả mẫu kiến trúc thường dùng khung nhìn
- a. Dòng dữ liệu
 - b. Module
 - c. Tiến trình
 - d. **Tất cả các mục trên**
201. Trong tích hợp module, gom cụm (cluster) được dùng trong
- a. **Tích hợp từ dưới lên**
 - b. Tích hợp big-bang
 - c. Tích hợp từ trên xuống
 - d. Tích hợp tăng vòng
202. Từ điển dữ liệu chứa những mô tả của mỗi
- a. Mục cấu hình phần mềm
 - b. **Đối tượng dữ liệu phần mềm**
 - c. Biểu đồ phần mềm
 - d. Hệ thống ký hiệu phần mềm
203. Use-cases là một kịch bản mà mô tả
- a. **Phần mềm thực hiện như thế nào khi được dùng trong một tình huống cho trước**
 - b. Những công cụ CASE sẽ được dùng như thế nào để xây dựng hệ thống
 - c. Kế hoạch xây dựng cho sản phẩm phần mềm
 - d. Những test-case cho sản phẩm phần mềm
204. Vấn đề nào sau đây liên quan chính đến pha thiết kế?

- a. Khả thi
- b. Dữ liệu**
- c. Tất cả các mục
- d. Phạm vi dự án

205. Xét đường độc lập cơ bản, nếu có 7 node phân nhánh thì ta có số đường thực thi cơ bản độc lập là

- a. 8**
- b. 7
- c. 9
- d. 6

1Dạng kiểm thử nào dùng kỹ thuật hộp trắng (white box test)

Kiểm thử hồi quy (regression test)

Kiểm thử nghiệm thu (acceptance test)

Kiểm thử hệ thống (system test)

Tất cả đều đúng

Câu 2 Phát biểu nào là sai khi nói về bản chất của phần mềm

- A) Có thể là sản phẩm theo đơn đặt hàng
- B) Là một sản phẩm công nghiệp
- C) Là sản phẩm có thể thực thi
- D) Không thực sự là sản phẩm**

Câu 14 Mật độ lỗi (defect density) dùng để đo lường

- A) Chất lượng sản phẩm cuối**
- B) Dự án phần mềm
- C) Quá trình sản xuất
- D) Chất lượng bảo trì

Câu 14 Mô hình nào dùng công cụ mạnh và thành phần tái sử dụng nhiều nhất?

- A) Mô hình xoắn ốc
- B) Mô hình RAD**
- C) Mô hình tăng dần
- D) Mô hình thác nước

Câu 14 Nguyên lý Pareto được áp dụng trong kiểm thử được phát biểu như sau:

- A) 80% lỗi trong chương trình thường do 20% bug gây ra**
- B) 20% lỗi trong chương trình thường do 80% bug gây ra
- C) Chi phí sửa lỗi ở giai đoạn thu nhận yêu cầu chỉ bằng 1/5 chi phí sửa lỗi ở

giai đoạn cuối

- D) 60% lỗi được tìm thấy trong giai đoạn kiểm thử đơn vị

Câu 14 Độ đo mức độ bảo trì

A) Số vấn đề giải quyết trong tháng /tổng số vấn đề phát sinh trong tháng

- B) Số lần bảo trì vượt quá tiêu chuẩn thời gian /tổng số lần bảo trì
- C) Số lần bảo trì sai sót /tổng số lần bảo trì
- D) Thời gian trung bình của một lần bảo trì

Câu 14 Mô tả nào sau đây có mức trừu tượng cao nhất:

- A) Kiến trúc hệ thống**
- B) Chi tiết các thành phần
- C) Các bảng dữ liệu và ràng buộc
- D) Mô tả một chức năng phần mềm

Câu 14 Phát biểu nào sau đây là sai khi nói về thiết kế

- A) Thiết kế không là code, code không là thiết kế
- B) Thiết kế phải được đánh giá chất lượng khi nó đang được tạo không phải khi nó có vấn đề
- C) Mô hình thiết kế cung cấp chi tiết về kiến trúc (architecture), Giao diện (interfaces) và thành phần (component) cần thiết để cài đặt phần mềm
- D) **Thiết kế phải chỉ ra được hệ thống thực thi như thế nào, các yêu cầu được hiện thực hóa ra sao**

Câu 14 Mức độ một module kết nối với các module khác chỉ tới

- A) **Tính liên kết (coupling)**
- B) Tính kết dính (cohesion)
- C) Chỉ đến chi phí tích hợp
- D) Chỉ đến chi phí phát triển

Câu 1 Phát biểu nào là hợp lý nhất khi nói về mô hình phát triển phần mềm tuần tự tuyến tính

- A) Một mô hình cũ phổ biến mà bây giờ hiếm khi dùng nữa
- B) Hướng tốt nhất để dùng cho những dự án với những nhóm phát triển lớn
- C) **Một hướng hợp lý khi những yêu cầu được xác định rõ**
- D) Một hướng tốt khi cần tạo nhanh một chương trình thực thi

Câu 15 Các đặc tính của mô hình tiến hóa

- A) Thường dùng prototype
- B) Bản chất lặp
- C) Dễ dàng điều tiết những biến đổi yêu cầu sản phẩm
- D) **Tất cả các mục**

Câu 16 Mô hình phát triển phần mềm dựa trên mẫu thử (prototype) là

- A) Một phương pháp thích hợp được sử dụng khi các yêu cầu đã được xác định rõ ràng
- B) Phương pháp tốt nhất được sử dụng trong các dự án có nhiều thành viên
- C) **Một phương pháp hữu ích khi khách hàng không thể xác định yêu cầu một cách rõ ràng**
- D) Một mô hình rất rủi ro, khó đưa ra được một sản phẩm tốt

Câu 17 Yêu cầu nào sau đây không phải là yêu cầu chức năng

- A) **Bảo mật**
- B) Các chi tiết về dữ liệu mà được tổ chức trong hệ thống
- C) Những mô tả về qui trình mà hệ thống được yêu cầu xử lý
- D) Các báo cáo kết xuất

Câu 18 Chọn lựa nào sau đây mô tả một yêu cầu chức năng?

- A) Hệ thống phải có khả năng trả lời tất cả các truy vấn trong 5 giây
- B) Các người sử dụng của hệ thống sẽ gây ra ít lỗi hơn 50% so với hệ thống hiện tại
- C) **Hệ thống phải cho phép những người sử dụng nhập vào các chi tiết của các chiến dịch quảng cáo**

D) Hàng tháng, báo cáo phải nộp lên giám đốc trước ngày 5 của tháng sau đó

Câu 19 Ví dụ nào sau đây không phải là yêu cầu phi chức năng?

- A) Phân chia ổ đĩa dữ liệu
- B) Các yêu cầu xử lý
- C) **Nội dung của các báo cáo in ra theo yêu cầu của hệ thống**
- D) Tất cả các mục

Câu 20 Chọn lựa nào sau đây mô tả một yêu cầu phi chức năng?

- A) **Hệ thống phải có khả năng lưu trữ ban đầu là 500MB dữ liệu, mỗi năm tăng lên 100MB**
- B) Hệ thống phải phát sinh ra một báo cáo về tất cả các chiến dịch quảng cáo cho một khách hàng cụ thể
- C) Hệ thống phải cho phép những người sử dụng nhập vào chi tiết các khách hàng
- D) Tất cả các Câu đều đúng

Câu 21 Trong lược đồ use case sau, phát biểu nào là sai

- A) **“Kiểm tra ngân quỹ chiến dịch” là use case cơ bản**
- B) “Kiểm tra ngân quỹ chiến dịch” là use case mở rộng được khởi động từ use case “In tóm tắt chiến dịch”
- C) “In tóm tắt chiến dịch” là use case cơ bản
- D) Tất cả đều sai

22. Nguyên tắc kiểm thử nào sau đây là sai:

Phải lên kế hoạch kiểm thử sớm ngay trong giai đoạn phân tích hệ thống

Có thể thực hiện kiểm thử được toàn bộ mọi trường hợp có thể có của hệ thống

Để hiệu quả, kiểm thử nên được thực hiện bởi một đội kiểm thử

Tuân theo nguyên tắc Pareto

22. Kỹ thuật gì nên dùng cho kiểm thử đơn vị:

- a. **Kỹ thuật hộp trắng**
- b. Kỹ thuật hộp đen
- c. Cả hai kỹ thuật hộp đen và trắng
- d. Kỹ thuật hồi quy (regression)

23. Dạng kiểm thử nào sau đây không thuộc kiểm thử hộp trắng:

- a. Kiểm thử điều kiện (Condition testing)
- b. Kiểm thử dòng dữ liệu (Data flow testing)
- c. **Kiểm thử vòng lặp (Loop testing)**
- d. Phân hoạch lớp tương đương (equivalent class partition)

24. Dạng kiểm thử nào sau đây không thuộc kiểm thử hộp đen:

- a. **Kiểm thử điều kiện (Condition testing)**
- b. Phân tích giá trị biên (boundary value analysis)
- c. Kiểm thử chuyển đổi trạng thái (State Transition Testing)
- d. Đoán lỗi (Error Guessing)

25. Xét đoạn mã giả sau:

```
if (input is in AllowedCharacterSet)
    if (input is a number)
        if (input >= 0)
            put input into positiveNumberList
        else
            put input into negativeNumberList
    else
        if (input is an alphabet)
            put input into alphabetList
        else
            put input into symbolList
else
    exception("Illegal character")
```

Cần tối thiểu bao nhiêu test case để độ bao phủ rẽ nhánh (branch coverage) là 100%

- a. 3
- b. 4
- c. 5
- d. 6

26. Xét chương trình tính phí cho việc thuê băng video như sau:

```
Float calcRentalFee(Tape[] tapes, Customer customer){
    float total = 0;
    for(int i = 0; i < tapes.length; i++){
        total += tapes[i].price;
    }
    if (tapes.length > 10){
        total *= .8;
    } else if(tapes.length > 5){
        total *= .9;
    }
    if(customer.isPremium()){
        total *= .9;
    }
    return total;
}
```

Nếu có các test case sau:

Test case 1 với tapes=[5,6,10,3,5,7,8] và Customer.isPremium = true

Test case 2 với tapes=[5,6,4,5,7,3,6,7,4,5,3,2] và Customer.isPremium = true

Test case 3 với tapes=[5,6,4,5] và Customer.isPremium = false

Thì tổ hợp các test case nào có độ bao phủ về lệnh (statement coverage) 100%

- a. Test case 1 và 2
- b. Test case 1 và 3
- c. Test case 2 và 3
- d. Cả 3 test case

27. Hoạt động sau đây thuộc loại bảo trì nào “Phát hiện sớm và sửa sai các khuyết điểm vừa mới phát hiện trước khi chúng trở thành các khuyết điểm chính”

- a. **Bảo trì sửa lỗi (Corrective maintenance)**
- b. Bảo trì thích nghi (Adaptive maintenance)

- c. Bảo trì hoàn chỉnh (Perfective maintenance)
- d. Bảo trì phòng tránh (Preventive maintenance)

28. Hoạt động sau đây thuộc loại bảo trì nào “Làm cho hệ thống tốt hơn, nhanh hơn, nhỏ hơn, tài liệu đầy đủ hơn”

- a. Bảo trì sửa lỗi (Corrective maintenance)
- b. Bảo trì thích nghi (Adaptive maintenance)
- c. **Bảo trì hoàn chỉnh (Perfective maintenance)**
- d. Bảo trì phòng tránh (Preventive maintenance)

29. Khi phần mềm bị lỗi, một trong những cách để khắc phục là “dùng miếng vá khẩn cấp (patching)”. Biện pháp này có tác dụng phụ gì?

- a. Tăng độ phức tạp của chương trình
- b. Tạo hiệu quả “ripple effect”
- c. Tăng độ bảo mật cho chương trình
- d. **Tất cả các chọn lựa**