

80 câu trắc nghiệm công nghệ phần mềm và đáp án

1. 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?

2. Ba giai đoạn tổng quát của công nghệ phần mềm

- definition, development, support
- what, how, where
- programming, debugging, maintenance
- analysis, design, testing

3. Mô hình phát triển ứng dụng nhanh

- Một cách gọi khác của mô hình phát triển dựa vào thành phần
- Một cách hữu dụng khi khách hàng không xác định yêu cầu rõ ràng
- Sự ráp nối tốc độ cao của mô hình tuần tự tuyến tính**
- Tất cả mục trên

4. Mô hình tiến trình phần mềm tiến hóa

- Bản chất lặp
- Đễ dàng điều tiết những biến đổi yêu cầu sản phẩm
- Nói chung không tạo ra những sản phẩm bỏ đi
- Tất cả các mục**

5. Mô hình phát triển phần mềm lặp lại tăng thêm

- Một hướng hợp lý khi yêu cầu được xác định rõ
- Một hướng tốt khi cần tạo nhanh một sản phẩm thực thi lõi**
- Một hướng tốt nhất dùng cho những dự án có những nhóm phát triển lớn
- Một mô hình cách mạng không nhưng không được dùng cho sản phẩm thương mại

6. Mô hình phát triển phần mềm xoắn ốc

- Kết thúc với việc xuất xưởng sản phẩm phần mềm
- Nhiều hỗn độn hơn với mô hình gia tăng

- s. Bao gồm việc đánh giá những rủi ro phần mềm trong mỗi vòng lặp
- t. Tất cả điều trên

7. Mô hình phát triển dựa vào thành phần

- u. Chỉ phù hợp cho thiết kế phần cứng máy tính
- v. Không thể hỗ trợ phát triển những thành phần sử dụng lại
- w. Dựa vào những kỹ thuật hỗ trợ đối tượng
- x. 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 giả lập hệ thống

- 13. 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**
- 14. 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
- 15. Mẫu mô hình hệ thống chứa thành phần**
- Input
 - Output
 - Giao diện người dùng
 - Tất cả mục trên**
- 16. 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
- 17. 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
- 18. Ai là người không thích hợp để tham dự vào nhóm FAST (facilitated application specification techniques)**
- Kỹ sư phần cứng và phần mềm
 - Đại diện nhà sản xuất
 - Đại diện thị trường
 - Nhân viên tài chính cao cấp**
- 19. Những yêu cầu nào được quan tâm suốt QFD (quality function deployment)**
- 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*
- Chức năng và hành vi
 - Giải thuật và cấu trúc dữ liệu
 - Kiến trúc và cấu trúc
 - Tính tin cậy và tính sử dụng
26. *Trong ngữ cảnh của phân tích yêu cầu, hai loại phân tách vấn đề là*
- bottom-up và top-down
 - horizontal and vertical
 - subordinate và superordinate
 - Không có mục nào
27. *Khung nhìn (view) nào được quan tâm đầu tiên trong phân tích yêu cầu phần mềm*
- actor view
 - data view
 - essential view
 - implementation view
28. *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
29. *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
30. *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

- 31. 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
- 32. 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
- 33. Biểu đồ dịch chuyển trạng thái**
- Đưa ra hình ảnh về các đối tượng dữ liệu
 - Đưa ra hình ảnh chức năng biến đổi luồng dữ liệu
 - Chỉ ra hình ảnh dữ liệu được biến đổi như thế nào bởi hệ thống
 - 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**
- Tự điền dữ liệu
 - Biểu đồ dòng dữ liệu**
 - Biểu đồ quan hệ thực thể
 - Biểu đồ dịch chuyển trạng thái
- 35. 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
- 36. Từ điển dữ liệu chứa những mô tả của mỗi**
- Mục cấu hình phần mềm
 - Đối tượng dữ liệu phần mềm**
 - Biểu đồ phần mềm
 - Hệ thống ký hiệu phần mềm
- 37. Mô hình thiết kế không quan tâm tới**
- Kiến trúc
 - Dữ liệu
 - Giao diện
 - Phạm vi dự án**

- 38. Sự quan trọng của thiết kế phần mềm có thể được tóm tắt bằng từ đơn**
- Accuracy
 - Complexity
 - Efficiency
 - Quality**
- 39. Một đặc trưng của thiết kế tốt là**
- Cho thấy sự liên kết mạnh giữa các module
 - Thực hiện tất cả yêu cầu trong phân tích**
 - Bao gồm những test case cho tất cả thành phần
 - Kết hợp mã nguồn nhằm mục đích mô tả
- 40. 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
- 41. Loại trừ tượng nào được dùng trong thiết kế phần mềm**
- Điều khiển
 - Dữ liệu
 - Thủ tục
 - Tất cả mục trên**
- 42. Loại mô hình nào không được có trong kiến trúc phần mềm**
- Dữ liệu**
 - Động
 - Xử lý
 - Cấu trúc
- 43. 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

44. **Thủ tục phần mềm tập trung vào**
- Cấp bậc điều khiển trong một cảm nhận trừu tượng hơn
 - Xử lý chi tiết của mỗi module riêng biệt**
 - Xử lý chi tiết của mỗi tập module
 - 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**
- 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
 - 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**
- 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
47. **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
48. **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
49. **Mẫu kiến trúc nhấn mạnh tới những thành phần**
- Ràng buộc
 - Tập hợp những thành phần
 - Mô hình ngữ nghĩa
 - 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**
- Giải thuật phức tạp
 - Đặc trưng và ràng buộc**
 - Điều khiển và dữ liệu
 - Những mẫu thiết kế
51. **Tiêu chuẩn đánh giá chất lượng của một thiết kế kiến trúc phải dựa vào**
- Tính truy cập và tính tin cậy của hệ thống
 - Dữ liệu và điều khiển của hệ thống**
 - Tính chức năng của hệ thống
 - 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**
- Dòng dữ liệu
 - Module
 - Tiến trình
 - 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**
- Liên kết thấp
 - Module hóa tốt
 - Luồng giao dịch (transaction)
 - 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**
- Liên kết thấp
 - Module hóa tốt
 - Luồng giao dịch (transaction)**
 - 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**
- Sơ đồ quan hệ - thực thể
 - Từ điển dữ liệu
 - Mô tả việc xử lý cho mỗi module**
 - 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**
- 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**
57. **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**
58. **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**
59. **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
60. **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**
61. **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
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**
- 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
70. **Những độ đo phức tạp vòng (cyclomatic complexity metric) cung cấp cho** người thiết kế thông tin về số
- Chu kỳ trong chương trình
 - Số lỗi trong chương trình
 - Những đường logic độc lập trong chương trình
 - 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**
- Dựa vào kiểm thử đường cơ bản
 - Thử thách điều kiện logic trong module phần mềm
 - Chọn những đường dẫn kiểm tra dựa vào những vị trí và dùng những biến
 - 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**
- Dựa vào kiểm thử đường cơ bản
 - Thử thách điều kiện logic trong module phần mềm
 - Chọn những đường dẫn kiểm tra dựa vào những vị trí và dùng những biến
 - 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**
- Dựa vào kiểm thử đường cơ bản
 - Thử thách điều kiện logic trong module phần mềm
 - Chọn những đường dẫn kiểm tra dựa vào những vị trí và dùng những biến
 - Tập trung vào việc kiểm thử việc giá trị những cấu trúc lặp
74. **Kiểm thử Black-box cố gắng tìm ra những lỗi**
- Chức năng không đầy đủ hay không đúng
 - Những lỗi giao diện
 - Những lỗi thực thi
 - Tất cả mục trên

- 75. Lý do tốt nhất cho việc dùng nhóm kiểm tra phần mềm độc lập là**
- Những người phát triển phần mềm không cần làm bất kỳ kiểm thử nào
 - Những người lạ sẽ kiểm phần mềm rất chặt
 - Những người kiểm thử không được đánh giá tới dự án cho đến khi kiểm thử bắt đầu
 - 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**
- Đưa ra những xem xét kỹ thuật hình thức ưu tiên trước khi kiểm thử
 - Chỉ rõ những yêu cầu trong theo một cách thức có thể định lượng
 - Quan tâm tới việc sử dụng những nhóm kiểm thử độc lập
 - Tất cả mục trên**
- 77. Kiểm thử tích hợp Top-down có thuận lợi chính là**
- Những module mức thấp không bao giờ cần kiểm thử
 - Những điểm quyết định chính được kiểm thử sớm**
 - Không có những stub cần phải viết
 - Không có mục nào
- 78. Kiểm thử tích hợp bottom-up có những thuận lợi chính**
- Những điểm quyết định chính được kiểm thử sớm
 - Không có những driver cần được viết
 - Không có những stub (nhánh) cần phải viết**
 - Không đòi hỏi kiểm thử hồi quy (regression)
- 79. Hướng debug**
- Backtracking
 - Brute force
 - Sự loại trừ nguyên nhân
 - Tất cả các mục**
- 80. Những kiểm tra chấp nhận thường được đưa ra bởi**
- Người phát triển
 - Những người dùng cuối**
 - Nhóm kiểm thử
 - Những kỹ sư hệ thống