PHƯƠNG PHÁP PHÁT TRIỂN PHẦN MỀM

**Bài 1: ICONIX Process tập trung vao giai đoạn nao trong vòng đoi phat triển phần mềm?**

\*A. Phân tích và thiết kế

B. Lập kế hoạch dự án

C. Quản lý dự án

D. Kiểm thử phần mềm

**Bài 2: Quy trình ICONIX giúp thu hẹp khoảng cách giữa lý thuyết và thực tế như thế nào?**

A.Chuyển đổi nhanh từ yêu cầu thành mã nguồn mà không qua phân tích

B. Tập trung vào thiết kế trực tiếp từ mã nguồn

\*C. Liên kết cac use case với các đối tượng trong hệ thống

D. Phát triển hệ thống dựa trên mô hình UML đầy đủ

**Câu3: Yêu cầu chức năng trả lời cho câu hỏi nào?**

A. Ai sẽ là người sử dụng hệ thống?

B. Hệ thống được triển khai như thế nào?

\*C. Hệ thống có thể làm gì?

D. Làm thế nào để thiết kế hệ thống?

**Câu 4: Quy trình ICONIX cố gắng giải quyết vấn đề gì trước đây?**

\*A. Cồng kềnh và nhiều thứ không cần thiết trong dự án

thực tế

B. Thiếu tài liệu mô tả hệ thống

C. Thiếu mô tả chức năng của mã nguồn

D. Thiếu sự chi tiết trong các mô hình

**Câu 5: Sự khác biệt chính giữa yêu cầu chức năng và yêu cầu hành vi trong quy trình ICONIX là gì?**

A. Yêu cầu chức năng mô tả chi tiết các lớp đối tượng,còn yêu cầu hành vi mô tả dữ liệu hệ thống

B. Cá hai đều là các bước đầu tiên của qua trình phân

tích hệ thống

C. Yêu cầu chức năng miêu tả cách hệ thống hoạt

động, còn yêu cầu hành vi miêu tả giao diện người

dùng

\*D. Yêu cầu chức năng mô ta những gì hệ thống cần

làm, còn yêu cầu hành vi mô tả cách hệ thống và

người dùng tương tác

**Câu 6: Trong quy trình ICONIX, hoạt động nao giup làm ro và loại bỏ sự mơ hồ trong yêu cầu hệ thống?**

A. Vẽ sơ đồ lớp

B. Kiểm thử đơn vị

\*C. Phân tích độ bền (robustness analysis)

D. Viết mã nguồn

**Câu 7: Trong quy trình ICONIX, mô hình miền có vai trò gì?**

\*A. Định nghĩa phạm vi và tạo nền tảng cho việc xây

dựng use case

B. Giúp xác định các yêu cầu không chức năng

C. Định nghĩa cách hệ thống sẽ triển khai

D. Đảm bảo tính khả dụng của hệ thống

**Câu 8: Việc sử dụng sơ đồ trình tự (sequence diagram) trong quy trình ICONIX nhằm mục đích gì?**

A. Để phân tích tính bảo mật của hệ thống

\*B. Để phân bổ hành vi cho các lớp đối tượng

C. Để phân tích tính khả thi của dự án

D. Để xác định các yêu cầu không chức năng

**Câu 9 :Trong quy trình ICONIX, tại sao cần thực hiện đánh giá thiết kế sơ bộ (Preliminary Design Review)?**

A. Để xác định tất ca các lỗi trong mã nguồn

\*B. Để đảm bảo mô hình miền và use case khớp nhau

C. Để kiểm tra tính chính xác cua mô hình dữ liệu

D. Để kiểm thử toàn bộ hệ thống

**Câu 10 :Làm thế nào để một sơ đồ độ bền giúp liên kết use case với mô hình đối tượng?**

A. Xác định các lỗi tiềm ẩn trong thiết kế

B. Liệt kê tất ca các yêu cầu chức năng

\*C.Xác định các đối tượng miền và các thông điệp truyền giữa chúng

D. Tạo ra các đối tượng giao diện (GUI)

**Câu 11: Khi cần kiểm tra việc tích hợp hệ thống dựa trên các use case trong ICONIX Process, ta nên bat đầu từ đâu?**

A. Kiểm tra cơ sở dữ liệu

B. Kiểm tra giao diện người dùng

C. Viết mã nguồn cho tất cả các chức năng

\*D. Kiểm tra kịch bản chính và các kịch bản thay thế

**Câu 12: Sau khi hoàn thành giai đoạn phân tích và thiết kế sơ bộ, nếu phát hiện sự khác biệt giữa use case và sơ đồ lớp, ta sẽ xử lý như thế nào?**

A. Kiểm tra mã nguồn

B. Bó qua sự khác biệt

C. Thực hiện kiểm thử đơn vị

\*D. Cập nhật sơ đồ lớp và use case để đảm bảo chúng

khớp với nhau

**Câu 13: Khi thực hiện sơ đồ trình tự, điều gì cần được xem xét đầu tiên để đảm bảo thiết kế đúng?**

A. Các bảng cơ sở dữ liệu

B. Các yêu cầu phi chức năng

C. Sơ đồ lớp

\*D. Các đối tượng biên và thực thể từ sơ đồ độ bền

**Câu 14: Tại sao quy trình ICONIX được coi là phù hợp với các dự án theo phương pháp phát triển Agile?**

A. Nó tập trung vao việc phát triển toàn bộ hệ thống ngay từ đầu

B. Nó yêu cầu một kế hoạch chi tiết và nghiêm ngặt từ đầu đến cuối

\*C. Nó cho phép lặp lại và phản hồi nhanh chóng dựa trên từng use case

D. Nó không cần mô hình hóa các yêu cầu

**Câu 15:Trong quá trình kiểm thử đơn vị, các bài kiểm thử được viết dựa trên:**

A. Kịch bản chính từ sơ đồ trình tự

B. Sơ đồ lớp

\*C. Use case và các controllers từ sơ đồ độ bền

D. Mô hình miền

**Câu 16:Quy trình ICONIX bao gồm bao nhiêu sơ đồ UML cốt lõi?**

A. 2

B. 6

\*C. 4

D. 14

**Câu 17: Câu hỏi chính mà quy trình ICONIX muốn trả lời là gì?**

\*A. Làm the nao đe có the đi từ use case đến ma nguồn

B. Làm thế nao để lập trình hướng đối tượng

C. Làm thế nào để đi từ người dùng đến sản phẩm

D. Làm the nao để sử dụng được hết các loại sơ đồ UML

**Câu 18: Đối với quy trình ICONIX, bước nào là quan trọng nhất để đảm bảo yêu cầu và thiết kế đồng nhất với nhau?**

A. Kiểm thử tích hợp

B. Kiểm thử đơn vị

C. Viết mã nguồn

\*D. Đánh giá thiết kế sơ bộ (PDR)

**Câu 19: Trong quy trình ICONIX, các yêu cầu hành vi được mô tả như thế nào?**

A. Thông qua mô hình lớp

B. Thông qua các bảng cơ sở dữ liệu

C. Thông qua mô hình dữ liệu

\*D. Thông qua use case mô tả cách người dùng và hệ thống tương tác

**Câu 20: Làm the nao để một sơ đồ độ bền giup liên kết use case với mô hình đối tượng?**

A. Xác định các lỗi tiềm ẩn trong thiết kế

B. Liệt kê tất cả các yêu cầu chức năng

C. Tạo ra các đối tượng giao diện (GUI)

\*D.Xác định các đối tượng miền và các thông điệp truyền giữa chúng

**Câu 21: Việc điều chỉnh mô hình miền trong ICONIX xảy ra khi nào?**

\*A. Khi phat hien cac đoi tượng mới trong qua trình thiết kế

B. Sau khi kiểm thử tích hợp

C. Trong quá trình kiểm thử đơn vị

D. Chỉ sau khi viết mã

**Câu22: Khi nao ta cần thực hiện đanh gia thiết kế chi tiết (Critical Design Review) trong quy trình ICONIX ?**

A. Sau khi kiểm thử đơn vị

B. Trước khi thiết kế sơ đồ lớp

\*C. Sau khi hoan thành sơ đồ trình tự và cập nhật mô hình tĩnh

D. Sau khi viết mã nguồn

**Câu 23: ICONIX Process là phương pháp tiếp cận nào?**

Phương pháp tuần tự

\*B. Phương pháp tối giản nhưng hoạt động

Phương pháp xoắn ốc

Phương pháp nhanh chóng

**Câu 24: Khi cần liên kết giua các use case và các đối tượng trong hệ thống, ta sử dụng sơ đồ nào trong quy trình ICONIX?**

A. Sơ đồ use case

\*B. Sơ đồ độ bền (robustness diagram)

C. Sơ đồ trình tự (sequence diagram)

D. Sơ đồ miền (domain model)

**Câu 25:Ta cần làm gì để đảm bảo rằng quá trình phân tích trong quy trình ICONIX tạo ra thiết kế có thể thực hiện được?**

A. Viết mã nguồn ngay lập tức

B. Phát triển sơ đồ lớp

C. Thực hiện phân tích độ bền (robustness analysis) để xác định các đối tượng cần thiết

D. Kiểm thử toàn bộ hệ thống

**Câu 26: Trong qua trình phát triển phần mềm, nếu có sự thay đổi về yêu cầu, bước nào trong ICONIX Process có thể giúp thích ứng với thay đổi đó?**

A. Cập nhật sơ đồ trình tự và thiết kế chi tiết

B. Thay đổi mô hình lớp

C. Viết mã nguồn mới

D. Kiểm thử toàn bộ hệ thống

**Câu 27: Đầu vào của quy trình ICONX là sơ đồ nào?**

A. Sequence diagram

\*B. Use case diagram

C. Robustness diagram

D. Class diagram

**Câu 28: Thông tin nào trong sơ đồ trình tự được dùng để bổ sung vào sơ đồ lớp**

\*A. Thông điệp

B. Đối tượng biên

C. Tác nhân

D. Đối tượng thực thể

**Câu 29: Mục tiêu chính của buổi đánh giá thiết kế quan trọng (CDR) là gì?**

A. Đánh giá các yêu cầu hệ thốn

B. Xác định các sơ đồ lớp

C. Kiểm thử mã nguồ

\*D. Kiểm tra tính nhất quán giữa thiết kế và yêu cầu

**Bài 2**

**Câu 1: Mô hình miền (domain model) trong quy trình ICONIX là gì?**

A. Là một bản thiết kế cơ sở dữ liệu

\*B. Là “từ điển” thuật ngữ trong dự án

C. Là sơ đồ quy trình nghiệp vụ

D. Là một bảng dữ liệu quan hệ

**Câu 2: Mô hình miền khác gì so với bảng thuật ngữ dự án (project glossary)?**

A. Mô hình miền chỉ chứa thông tin về các lớp (classes) trong dự án.

B. Mô hình miền chỉ sử dụng trong giai đoạn kiểm thử dự án.

\*C. Mô hình miền là một biểu đồ được minh họa, hiển thị mối quan hệ giữa các thuật ngữ.

D. Mô hình miền không có mối quan hệ với bảng thuật ngữ dự án.

**Câu 3: Mô hình miền là cơ sở đầu tiên cho việc xây dựng sơ đồ nào trong quy trình ICONIX?**

A. Sơ đồ mạnh mẽ

\*B. Sơ đồ use case

C. Sơ đồ hoạt động

D. Sơ đồ trình tự

**Câu 4: Điều nào sau đây không nên thực hiện khi xây dựng mô hình miền?**

\*A. Đưa các lớp giao diện người dùng vào mô hình miền.

B. Sắp xếp các lớp dựa trên các khái niệm chính trong miền vấn đề.

C. Sử dụng mô hình miền như một bảng thuật ngữ của dự án.

D. Giới hạn thời gian tạo mô hình miền ban đầu trong vài giờ.

**Câu 5: Tại sao nên bắt đầu với mô hình miền (domain model) trước khi viết use case?**

A. Vì use case không liên quan đến mô hình miền.

B. Vì use case nên được viết trước để định nghĩa chi tiết.

\*C. Vì nó giúp ràng buộc use case vào thực tế và hệ thống sẽ được thiết kế.

D. Vì mô hình miền phức tạp hơn và nên được thực hiện trước.

**Câu 6:** Điều nào sau đây không nên thực hiện khi xây dựng mô hình miền?

\*A. Sử dụng quan hệ tổng quát hóa (generalization) để đơn giản hóa mối quan hệ.

B. Thêm nhiều lớp hơn để giải thích.

C. Bỏ bớt các lớp.

D. Giữ nguyên và tiếp tục.

**Câu 7: Nếu phát hiện mô hình miền thiếu một số đối tượng quan trọng, ta nên làm gì?**

A. Tạo một sơ đồ mới.

\*B. Cập nhật mô hình miền để bao gồm các đối tượng bị thiếu.

C. Bỏ qua những đối tượng này.

D. Xây dựng lại mô hình từ đầu.

**Câu 8: Mối quan hệ nào dưới đây là ví dụ về quan hệ "is-a"?**

A. Mục Đơn Hàng thuộc Giỏ Hàng.

B. Quyển Sách có Mục Lục.

\*C. Con Mèo là một Loài Động Vật.

D. Người Dùng có Tài Khoản.

**Câu 9: Quan hệ "has-a" trong mô hình miền là loại quan hệ gì?**

A. Quan hệ một-nhiều.

\*B. Tập hợp (aggregation).

C. Tổng quát hóa.

D. Kế thừa.

**Câu 10: Vai trò của mô hình miền trong quá trình phát triển hệ thống là gì?**

\*A. Để đảm bảo giao tiếp rõ ràng, nhất quán giữa các thành viên dự án.

B. Để mô hình hóa cơ sở dữ liệu.

C. Để tối ưu hóa mã nguồn.

D. Để kiểm thử hệ thống.

**Câu 11: Nếu phát hiện hai thuật ngữ "Book" và "Books" trong mô hình miền, bước tiếp theo bạn nên làm gì?**

\*A. Xác định và loại bỏ sự trùng lặp, giữ lại một trong hai thuật ngữ.

B. Giữ lại cả hai thuật ngữ.

C. Tạo lớp mới cho mỗi thuật ngữ.

D. Thêm thuộc tính cho cả hai.

**Câu 12: Quá trình nào giúp phát hiện các đối tượng bị thiếu trong mô hình miền sau khi phiên tạo mô hình ban đầu hoàn thành?**

A. Triển khai hệ thống.

B. Phân tích mã nguồn.

\*C. Phân tích use case và sơ đồ độ bền (robustness diagrams).

D. Kiểm thử hệ thống.

**Câu 13: Lợi ích của việc tổ chức mô hình miền dựa trên các trừu tượng trong thế giới thực là gì?**

A. Giúp giảm bớt tài liệu yêu cầu.

\*B. Giúp mô hình bền vững hơn trước những thay đổi của yêu cầu phần mềm.

C. Giúp tăng tốc độ triển khai hệ thống.

D. Làm cho mô hình khó thay đổi hơn khi yêu cầu thay đổi.

**Câu 14: Mục tiêu chính của mô hình miền là gì?**

\*A. Giúp tạo ra một ngôn ngữ chung để truyền đạt thông tin rõ ràng giữa các thành viên dự án

B. Để xác định các yêu cầu phi chức năng

C. Để thiết kế cơ sở dữ liệu

D. Để xây dựng mô hình dữ liệu chính xác

**Câu 15: Mô hình miền đại diện cho phần nào của hệ thống?**

A. Mô tả các trường hợp sử dụng chi tiết.

B. Chỉ mô tả các giao diện người dùng.

\*C. Phần tĩnh mô tả cấu trúc của hệ thống.

D. Phần động mô tả hành vi của hệ thống.

**Câu 16: Trong mô hình miền, tại sao không nên nhầm lẫn giữa đối tượng và bảng cơ sở dữ liệu?**

\*A. Vì đối tượng đại diện cho một thể hiện duy nhất, trong khi bảng cơ sở dữ liệu đại diện cho một tập hợp.

B. Vì các đối tượng phức tạp hơn bảng dữ liệu

C. Vì bảng cơ sở dữ liệu không có quan hệ với các đối tượng

D. Vì bảng cơ sở dữ liệu chứa nhiều khóa ngoài

**Câu 17: Tại sao việc xây dựng mô hình miền ban đầu chỉ nên kéo dài khoảng 2 giờ?**

A. Vì thời gian giới hạn của dự án

B. Vì không cần mô hình lớp chi tiết

\*C. Vì mô hình miền sẽ được cập nhật liên tục trong quá trình dự án.

D. Vì chỉ có các yêu cầu chức năng quan trọng

**Câu 18: Lý do nên bắt đầu với mô hình miền thay vì use case là gì?**

A. Vì use case không cần phải rõ ràng ngay từ đầu

B. Vì mô hình miền dễ thực hiện hơn

\*C. Để đảm bảo use case được viết trong bối cảnh của mô hình đối tượng và tránh sự mơ hồ.

D. Để tiết kiệm thời gian

**Câu 19: Tại sao khi tạo mô hình miền, cần tập trung vào các đối tượng trong thế giới thực?**

A. Vì các đối tượng trong thế giới thực đơn giản hơn các đối tượng trong phần mềm.

\*B. Vì phần mềm nên được thiết kế sao cho giống với cách thế giới thực hoạt động, và thế giới thực thay đổi ít thường xuyên hơn so với yêu cầu phần mềm.

C. Vì nó giúp tránh sử dụng quá nhiều tài nguyên trong quá trình phát triển phần mềm.

D. Vì các đối tượng trong thế giới thực không ảnh hưởng đến kiến trúc phần mềm.

**Câu 20: Mối quan hệ nào sau đây là đúng khi nói về các mối quan hệ trong mô hình miền?**

A. Mối quan hệ giữa các lớp không quan trọng trong mô hình miền.

\*B. Mối quan hệ tổng hợp (has-a) thể hiện một đối tượng sở hữu một đối tượng khác và mối quan hệ tổng quát hóa (is-a) thể hiện sự kế thừa giữa các lớp.

C. Mối quan hệ tổng quát hóa (is-a) chỉ thể hiện một lớp cha có nhiều lớp con.

D. Tất cả các mối quan hệ trong mô hình miền đều là các mối quan hệ tuần tự.

**Câu 21: Ý nghĩa của các mối quan hệ "has-a" và "is-a" trong mô hình miền là gì?**

A. Chúng không có trong mô hình miền.

B. Chúng chỉ áp dụng cho các lớp giao diện (interfaces).

C. Chúng là các mối quan hệ giữa các thuộc tính của lớp.

.\*D. Chúng mô tả cách các lớp chứa các lớp khác và các lớp được kế thừa.

**Câu 22: Tại sao không nên kỳ vọng sơ đồ lớp cuối cùng sẽ khớp chính xác với mô hình miền?**

A. Vì sơ đồ lớp không quan trọng

B. Vì mô hình miền không liên quan đến sơ đồ lớp

.\*C. Vì mô hình miền đơn giản hơn và không chứa các chi tiết thiết kế cụ thể.

D. Vì sơ đồ lớp chỉ là mô hình tạm thời

**Câu 23: Điểm khác biệt chính giữa một đối tượng và một bảng cơ sở dữ liệu là gì?**

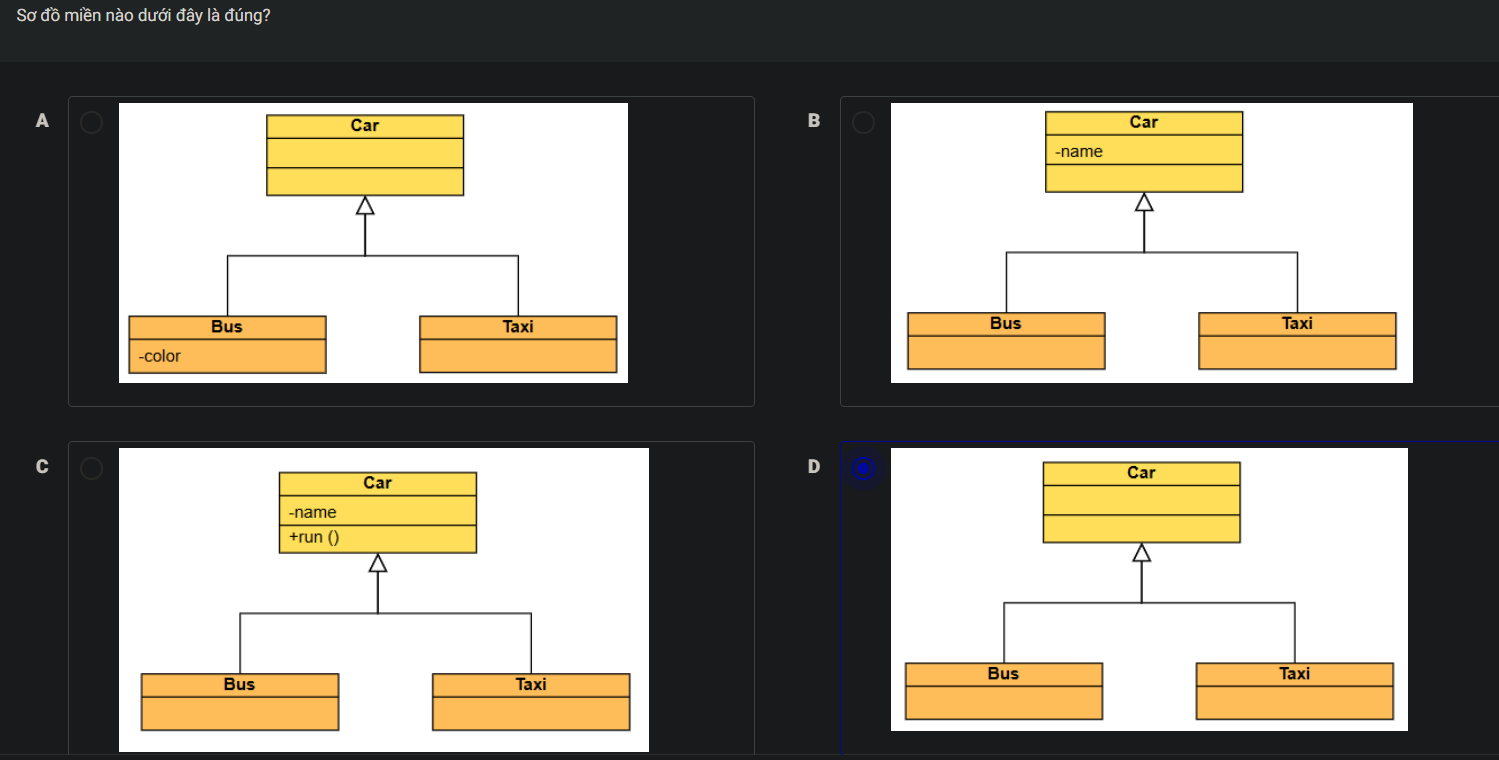
A. Bảng cơ sở dữ liệu không liên quan gì đến đối tượng trong mô hình miền.

B. Một đối tượng đại diện cho một tập hợp các bản ghi, trong khi bảng đại diện cho một bản ghi duy nhất.

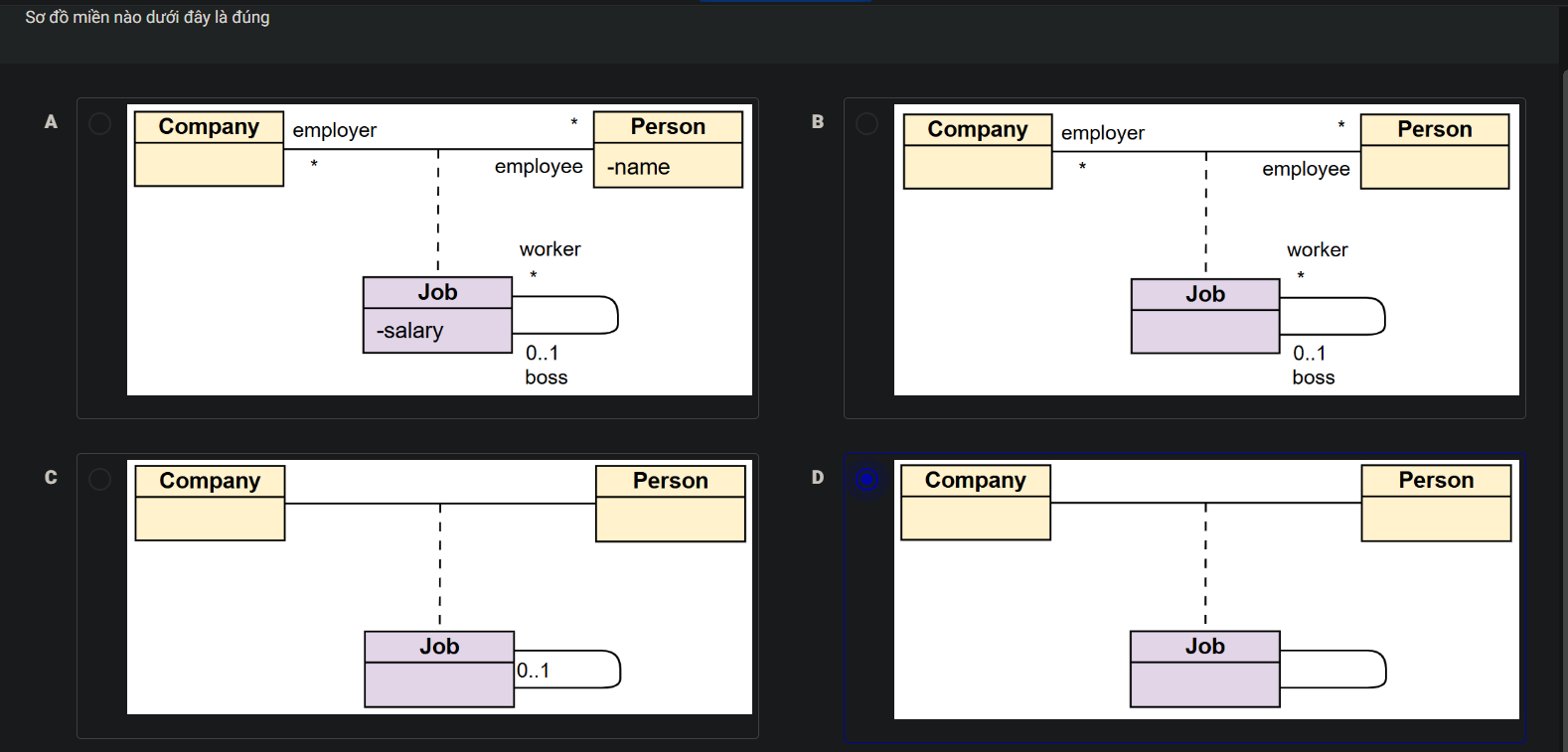
.\*C. Một đối tượng đại diện cho một thể hiện đơn lẻ, trong khi bảng cơ sở dữ liệu đại diện cho một tập hợp các bản ghi

D. Một đối tượng luôn chứa nhiều thuộc tính hơn bảng cơ sở dữ liệu.

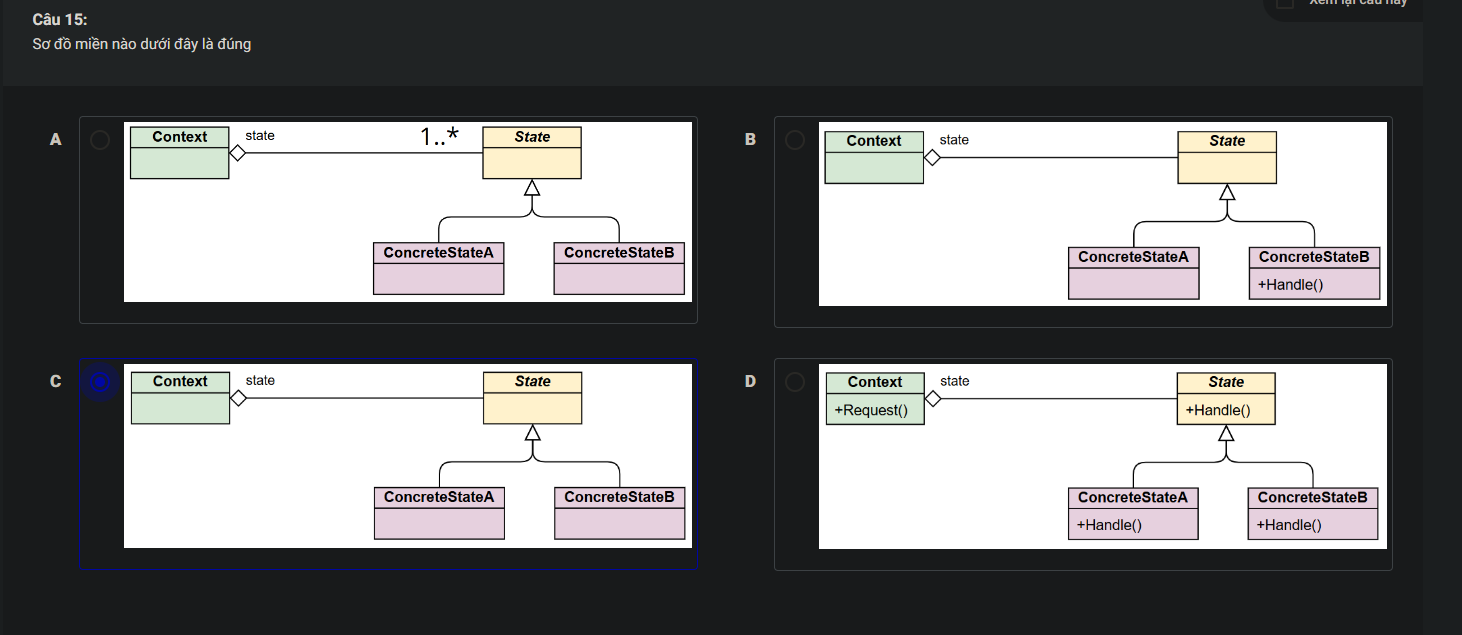
**Câu 24:**

****

**Câu 25:**

****

**Câu 26:**

****

**Câu 27 : Mục tiêu của phiên tạo mô hình miền ban đầu là gì?**

A. Tìm ra các lỗi phần mềm trong hệ thống.

B. Tạo ra mô hình hoàn chỉnh mà không cần sửa đổi sau này.

\*C. Xây dựng một phiên bản sơ bộ để có thể điều chỉnh trong quá trình làm việc.

D. Hoàn thiện mô hình miền ngay từ đầu..

**Câu 28: Nguyên tắc nào đúng khi nói về quan hệ giữa các đối tượng trong mô hình miền?**

A. Tránh sử dụng bất kỳ quan hệ nào để giữ mô hình đơn giản.

B. Không sử dụng quan hệ tổng quát hóa (is-a) và tổng hợp (has-a).

\*C. Sử dụng quan hệ tổng quát hóa (is-a) và tổng hợp (has-a) để thể hiện cách các đối tượng liên quan với nhau.

D. Chỉ sử dụng các quan hệ tuần tự để kết nối các đối tượng.

**Câu 29: Mô hình miền được minh họa như thế nào?**

\*A. Như một sơ đồ lớp (class diagram) đơn giản với các đường nối giữa các lớp.

B. Như một sơ đồ trình tự (sequence diagram).

C. Như một sơ đồ hoạt động (activity diagram).

D. Như một sơ đồ triển khai (deployment diagram).

**Câu 30: Mô hình miền hiển thị loại quan hệ nào giữa các lớp trong miền?**

A. Quan hệ tuần tự (sequential relationships).

B. Quan hệ phụ thuộc (dependency relationships).

\*C. Quan hệ tổng hợp (aggregation) và tổng quát hóa (generalization).

D. Quan hệ đồng thời (concurrent relationships).

**Bài 3**

**Câu 1: Use case giúp trả lời câu hỏi nào liên quan đến hệ thống?**

A. Cách hệ thống lưu trữ dữ liệu trong cơ sở dữ liệu.

B. Hệ thống sẽ được triển khai như thế nào?

C. Sự phân bổ nguồn lực phát triển trong dự án.

\* D. Người dùng của hệ thống đang cố gắng làm gì và trải nghiệm người dùng ra sao?

**Câu 2: Tại sao không nên thiết kế, lập trình, hoặc ước tính trực tiếp từ tài liệu yêu cầu chức năng (functional specification)?**

A. Vì tài liệu yêu cầu chức năng chỉ dành cho nhóm quản lý.

\* B. Vì thiết kế trực tiếp từ tài liệu yêu cầu chức năng giống như chơi trò “chọn số ngẫu nhiên”, cần có thêm bước khám phá để làm rõ yêu cầu.

C. Vì tài liệu yêu cầu chức năng thường có nhiều lỗi.

D. Vì tài liệu yêu cầu chức năng không liên quan đến quá trình thiết kế phần mềm.

**Câu 3: Điều gì làm cho yêu cầu chức năng (functional requirements) trở thành bước đầu tiên trong quá trình thu thập yêu cầu?**

A. Vì yêu cầu chức năng chỉ bao gồm các yêu cầu kỹ thuật.

\* B. Vì yêu cầu chức năng là tập hợp các yêu cầu ban đầu từ các bên liên quan, giúp xác định các yêu cầu hành vi cần làm rõ hơn.

C. Vì yêu cầu chức năng cung cấp một bản thiết kế chi tiết của hệ thống.

D. Vì yêu cầu chức năng thay thế hoàn toàn các trường hợp sử dụng.

**Câu 4: Yếu tố nào sau đây giúp ích đáng kể trong việc xác định use case?**

A. Phân tích cấu trúc cơ sở dữ liệu.

\* B. Tạo storyboards và nguyên mẫu (UI mockups, demo chức năng cơ bản).

C. Viết tài liệu quản lý dự án.

D. Thực hiện kiểm thử hiệu năng.

**Câu 5: Khi viết use case, điều gì cần được lưu ý ngoài kịch bản chính (sunny-day scenario)?**

A. Chỉ viết về cách hệ thống hoạt động trong điều kiện tối ưu.

B. Tập trung vào tài liệu kỹ thuật thay vì các tình huống sử dụng thực tế.

C. Chỉ cần tập trung vào kịch bản chính mà không cần các kịch bản khác.

\* D. Phân tích các kịch bản thay thế (rainy-day scenarios) để mô tả cách hệ thống phản ứng khi xảy ra lỗi hoặc khi người dùng sử dụng các tính năng ít phổ biến.

**Câu 6: Trong ngữ cảnh của use case, “rainy-day scenario” là gì?**

A. Một kịch bản mô tả cách người dùng sử dụng hệ thống trong điều kiện lý tưởng.

B. Một kịch bản mô tả giao diện người dùng.

C. Một kịch bản mô tả các tính năng hiệu năng của hệ thống.

\* D. Một kịch bản mô tả cách hệ thống hoạt động trong các tình huống không mong muốn hoặc khi người dùng sử dụng các tính năng ít phổ biến.

**Câu 7: Tại sao việc ghi nhận các "rainy-day scenarios" (kịch bản không mong muốn) lại quan trọng khi viết use case?**

\* A. Vì các kịch bản này cung cấp thông tin về cách hệ thống cần phản hồi khi có lỗi hoặc khi người dùng tương tác với các tính năng ít được sử dụng, đảm bảo hệ thống hoạt động toàn diện và ổn định.

B. Vì các kịch bản này không ảnh hưởng đến cách hệ thống hoạt động trong thực tế.

C. Vì các kịch bản này chỉ hữu ích trong giai đoạn kiểm thử hệ thống.

D. Vì các kịch bản này giúp xác định những tính năng không cần thiết trong hệ thống.

**Câu 8: Tại sao việc viết use case nên tập trung vào sự tương tác giữa người dùng và hệ thống?**

A. Để xác định các lỗi trong hệ thống.

\* B. Để mô tả rõ ràng các hành vi cần thiết mà hệ thống phải thực hiện.

C. Để tạo sơ đồ trạng thái.

D. Để đảm bảo tính an toàn của hệ thống.

**Câu 9: Tại sao việc viết use case theo cấu trúc "noun-verb-noun" lại quan trọng?**

\*A. Để tạo ra một thiết kế hướng đối tượng dễ dàng từ use case.

B. Để dễ dàng mô hình hóa cơ sở dữ liệu.

C. Để tạo sơ đồ lớp.

D. Để viết mã nguồn trực tiếp.

**Câu 10: Trong quá trình viết use case, nếu người dùng thực hiện một hành động không dự đoán trước, bạn sẽ làm gì?**

A. Thay đổi mô hình miền.

\* B. Thêm vào "rainy-day scenario" trong use case để mô tả hành vi này.

C. Yêu cầu người dùng không thực hiện hành vi đó.

D. Bỏ qua yêu cầu hành vi này.

**Câu 11: Nếu phát hiện ra một yêu cầu mới trong quá trình viết use case, bạn sẽ làm gì?**

A. Bỏ qua yêu cầu này.

\* B. Cập nhật use case để bao gồm yêu cầu mới.

C. Viết mã nguồn cho yêu cầu này.

D. Thay đổi toàn bộ thiết kế hệ thống.

**Câu 12: Nếu mô hình miền thay đổi trong quá trình phát triển, điều gì cần thực hiện với use case?**

A. Thay đổi mô hình lớp.

\* B. Cập nhật use case để phù hợp với mô hình miền mới.

C. Bỏ qua mô hình miền.

D. Tạo sơ đồ trình tự mới.

**Câu 13: Tại sao nên xây dựng mô hình miền ban đầu trước khi viết use case?**

A. Để tránh phải sửa đổi mô hình miền sau khi phân tích use case.

B. Để đảm bảo mô hình miền hoàn chỉnh ngay từ đầu.

\* C. Để thiết lập bối cảnh cho các thuật ngữ được sử dụng trong use case và cung cấp cơ sở cho việc phát triển use case.

D. Để mô tả chi tiết tất cả các tương tác giữa người dùng và hệ thống ngay từ đầu.

**Câu 14: Viết một đoạn văn bản mô tả một "sunny-day scenario" cho chức năng Đăng nhập trong hệ thống bán hàng trực tuyến, sử dụng cấu trúc "noun-verb-noun".**

A. Người dùng nhập mật khẩu và đăng nhập thành công.

B. Hệ thống hiển thị màn hình đăng nhập.

\* C. Người dùng nhập tên đăng nhập và mật khẩu, sau đó nhấn nút Đăng nhập. Hệ thống xác thực thông tin và đăng nhập người dùng.

D. Người dùng vào trang đăng nhập, kiểm tra thông tin và đăng nhập thành công.

**Câu 15: Mô tả một "rainy-day scenario" cho tình huống người dùng nhập sai mật khẩu.**

A. Hệ thống chấp nhận mọi đầu vào và cho phép người dùng đăng nhập.

B. Hệ thống chuyển người dùng đến trang chính.

\* C. Hệ thống hiển thị thông báo lỗi "Mật khẩu không đúng" và yêu cầu người dùng nhập lại mật khẩu.

D. Hệ thống không phản hồi gì và yêu cầu người dùng nhập lại thông tin.

**Câu 16: Use case là gì trong quy trình ICONIX?**

A. Một báo cáo tài chính liên quan đến chi phí phát triển.

B. Một sơ đồ thể hiện các mối quan hệ giữa các lớp trong hệ thống.

C. Một công cụ kiểm thử phần mềm tự động.

\*D. Một cách có cấu trúc để ghi lại các yêu cầu hành vi của hệ thống.

**Câu 17: Use case cung cấp điều gì cho quá trình thiết kế hệ thống?**

\*A. Một nền tảng để thiết kế và ước tính thời gian, công sức một cách đáng tin cậy.

B. Một cách để dự đoán lỗi trong phần mềm.

C. Một phương pháp để viết tài liệu dự án theo cách tổng quát.

D. Một danh sách các công cụ cần sử dụng trong dự án.

**Câu 18: Trong quá trình viết use case, điều gì cần được chú ý nhất?**

A. Xác định yêu cầu phi chức năng.

B. Tạo ra nhiều sơ đồ lớp.

C. Viết mã nguồn trực tiếp từ use case.

\*D. Sử dụng các thuật ngữ rõ ràng và nhất quán với mô hình miền.

**Câu 19: Khi mô hình miền và use case không khớp nhau, bạn nên làm gì?**

A. Kiểm tra lại mô hình dữ liệu.

\*B. Cập nhật mô hình miền hoặc use case để đảm bảo sự khớp nhau.

C. Làm việc lại với khách hàng để thu thập thêm yêu cầu.

D. Bỏ qua sự không khớp.

**Câu 20: Khi hệ thống gặp lỗi trong quá trình kiểm thử và không thực hiện đúng theo use case, bước tiếp theo là gì?**

A. Thay đổi yêu cầu ban đầu.

B. Xây dựng lại hệ thống từ đầu.

\*C. Chỉnh sửa mã nguồn để phản ánh đúng hành vi trong use case

D. Tạo thêm các sơ đồ trạng thái.

**Câu 21: Khi viết use case, tại sao cần tránh sử dụng câu bị động?**

A. Để dễ dàng viết mã nguồn.

B. Để giảm bớt sự phức tạp trong thiết kế.

\*C. Để đảm bảo rõ ràng ai đang thực hiện hành động và hệ thống phản hồi như thế nào.

D. Để mô tả hệ thống theo yêu cầu.

**Câu 22: Theo quy trình ICONIX, tại sao mô hình miền ban đầu thường được coi là không hoàn chỉnh?**

A. Vì tất cả các yêu cầu đều phải được xác định rõ ràng trước khi bắt đầu xây dựng mô hình miền.

\*B. Vì mô hình miền ban đầu chỉ là một bản sơ bộ và sẽ được cải thiện dần qua quá trình phân tích use case.

C. Vì mô hình miền ban đầu không liên quan đến use case.

D. Vì không thể bao gồm tất cả các chức năng trong mô hình đầu tiên.

**Câu 23: Hãy tạo một mô hình miền ban đầu cho một hệ thống bán sách trực tuyến. Bạn sẽ chọn những thực thể nào và tại sao?**

A. Chọn các thực thể liên quan đến hệ thống thanh toán và giao hàng, vì đó là phần quan trọng nhất trong hệ thống bán sách.  
\*B**.** Chọn các thực thể như Sách, Tác giả, và Giỏ hàng, vì đây là các thực thể chính của hệ thống bán sách trực tuyến và phù hợp với nhu cầu người dùng.  
C. Chọn các thực thể liên quan đến quản trị hệ thống, vì nó giúp đảm bảo hệ thống vận hành đúng cách.  
D. Chọn tất cả các thực thể có thể xuất hiện trong hệ thống, để không bỏ sót thông tin nào.

**Câu 24: Trong quá trình viết use case, bạn phát hiện hệ thống cần phản hồi khi người dùng nhập sai mật khẩu. Hãy mô tả một "rainy-day scenario" cho tình huống này.**

\*A. Hệ thống hiển thị thông báo lỗi "Mật khẩu không đúng" và yêu cầu người dùng nhập lại mật khẩu.

B. Hệ thống không phản hồi gì và yêu cầu người dùng nhập lại thông tin.

C. Hệ thống chấp nhận mọi đầu vào và cho phép người dùng đăng nhập.

D. Hệ thống chuyển người dùng đến trang chính.

Dưới đây là câu hỏi và đáp án theo định dạng bạn đã yêu cầu:

**Câu 25: Mục đích chính của use case là gì?**

A. Thể hiện cấu trúc dữ liệu của hệ thống.

\*B. Giúp nắm bắt các yêu cầu hành vi của hệ thống để tạo ra thiết kế.

C. Tìm ra cấu trúc tĩnh của hệ thống.

D. Tạo ra một danh sách các lỗi tiềm ẩn trong hệ thống.

**Câu 26 Mục tiêu của use case trong quy trình hướng đối tượng là gì?**

A. Để tạo một tài liệu mô tả toàn bộ quá trình phát triển.

\*B. Là bước đầu tiên trong việc thiết kế hướng đối tượng và để nhanh chóng đạt được mã chất lượng cao.

C. Để xác định chi tiết các cấu trúc dữ liệu trong cơ sở dữ liệu.

D. Để làm tài liệu trừu tượng về yêu cầu.

**Câu 27: Khi phát hiện một đối tượng không được mô tả trong use case nhưng có liên quan đến hệ thống, bạn sẽ làm gì?**

A. Bỏ qua đối tượng này.

B. Xóa bỏ đối tượng.

C. Chỉ thay đổi mô hình miền.

\*D. Thêm đối tượng vào use case và mô hình miền.

**Câu 28: Khi xây dựng use case, việc tổ chức use case với các tác nhân (actors) có mục đích gì?**

A. Để đảm bảo tính toàn vẹn của hệ thống.

\*B. Để làm rõ vai trò của từng người dùng và chức năng tương ứng họ thực hiện.

C. Để tối ưu hóa mã nguồn.

D. Để phân tích mô hình dữ liệu.

**Câu 29: Vai trò của use case trong việc bổ sung cho yêu cầu chức năng là gì?**

A. Để thay thế hoàn toàn tài liệu yêu cầu chức năng.

\*B. Để cung cấp một phương pháp có cấu trúc giúp ghi lại các yêu cầu hành vi, hỗ trợ quá trình thiết kế.

C. Để lập danh sách các lỗi tiềm ẩn trong hệ thống.

D. Để xác định các trường hợp thử nghiệm phần mềm.

**Câu 30: Điều gì xảy ra với mô hình miền khi các use case được phát triển?**

A. Use case sẽ thay thế hoàn toàn mô hình miền.

B. Mô hình miền sẽ giữ nguyên và không thay đổi.

C. Mô hình miền sẽ chỉ được sử dụng lại khi hệ thống hoàn thành.

\*D. Mô hình miền sẽ được cập nhật và cải thiện liên tục dựa trên thông tin và thay đổi từ use case.

**Bài 4**

**Question 1:Điều gì nên được sử dụng để bổ sung mô tả use case?**

A. Các bảng số liệu phân tích.

B. Các tài liệu kỹ thuật chi tiết.

C. Các bảng màu thiết kế giao diện.

\*D. Các nguyên mẫu GUI hoặc bản phác thảo màn hình.

**Question 2: Nguyên tắc nào là quan trọng nhất khi xây dựng mô hình miền?**

A. Đảm bảo rằng tất cả các đối tượng đều được triển khai chi tiết.

B. Bao gồm tất cả các khía cạnh kỹ thuật.

C. Chỉ sử dụng ngôn ngữ kỹ thuật.

\*D. Bao quát ít nhất 80% các khái niệm quan trọng từ miền vấn đề.

**Question 3:Một trong các bước đầu tiên khi đánh giá use case là gì?**

A. Chuyển các yêu cầu chức năng vào use case.

\*B. Loại bỏ mọi thứ nằm ngoài phạm vi.

C. Tạo sơ đồ giao diện người dùng.

D. Thêm các yêu cầu mới vào mô hình.

**Question 4: Khi mô tả hành vi người dùng và phản hồi của hệ thống, điều gì cần được tránh?**

\*A. Kết hợp ngôn ngữ bị động và chủ động.

B. Mô tả chi tiết từng bước.

C. Sử dụng ngôn ngữ cụ thể.

D. Liên kết các yêu cầu với giao diện người dùng.

**Question 5:Điều gì sẽ xảy ra nếu các yêu cầu chức năng bị lẫn vào văn bản mô tả use case?**

A. Tăng tính rõ ràng của tài liệu.

B. Đơn giản hóa quá trình kiểm thử.

\*C. Gây nhầm lẫn khi phân tích và thiết kế.

D. Tăng tính linh hoạt khi cập nhật yêu cầu.

**Question 6: Việc tổ chức các use case thành các gói giúp ích như thế nào trong quy trình phát triển?**

A. Đơn giản hóa việc liên kết các yêu cầu với các thành phần phần mềm.

B. Giảm chi phí phát triển.

\*C. Cung cấp cái nhìn tổng quan rõ ràng hơn về các chức năng của hệ thống.

D. Tăng tốc độ viết tài liệu.

**Question 7: Tại sao cần có các "kịch bản thay thế" trong mỗi use case?**

A. Để đáp ứng các yêu cầu cấp cao của khách hàng.

B. Để đảm bảo rằng tất cả các yêu cầu chức năng được thực hiện.

C. Để đơn giản hóa mô tả use case.

\*D. Để xử lý các tình huống đặc biệt hoặc lỗi tiềm ẩn trong hệ thống.

**Question 8: Việc xác định các liên kết "is-a" và "has-a" trong mô hình miền có vai trò gì?**

A. Đảm bảo rằng các yêu cầu được đáp ứng đầy đủ.

B. Loại bỏ nhu cầu tạo các lớp đối tượng.

\*C. Tăng tính chính xác trong việc mô tả các mối quan hệ giữa các đối tượng.

D. Làm giảm độ phức tạp của hệ thống.

**Question 9:Trong buổi đánh giá yêu cầu, tại sao cần tạo nguyên mẫu (prototype) cho giao diện người dùng?**

A. Để giảm chi phí phát triển.

B. Để tạo tài liệu thiết kế chi tiết hơn.

\*C. Để xác minh rằng các yêu cầu được thể hiện đúng trong thiết kế giao diện.

D. Để tăng số lượng các yêu cầu được xử lý.

**Question 10:Tại sao nên sử dụng các bước "Hành động của người dùng / Phản hồi của hệ thống" trong mô tả use case?**

A. Để tăng độ phức tạp của tài liệu.

B. Để giảm thời gian thực hiện kiểm thử.

\*C. Để làm rõ cách hệ thống tương tác với người dùng.

D. Để đảm bảo rằng tất cả các yêu cầu kỹ thuật đều được đáp ứng.

**Question 11: Vai trò của việc tạo các nguyên mẫu giao diện (prototypes) trong buổi đánh giá yêu cầu là gì?**

A. Để tập trung vào các yêu cầu trừu tượng hơn.

B. Để xây dựng các chức năng phức tạp hơn.

\*C. Để xác định các hành vi người dùng mà tài liệu chưa nêu rõ.

D. Để giảm chi phí phát triển phần mềm.

**Question 12:Việc chuyển đổi từ giọng bị động sang giọng chủ động trong mô tả use case có tác dụng gì?**

A. Giúp tài liệu trở nên dễ kiểm tra hơn trong giai đoạn kiểm thử.

B. Tăng tính chi tiết trong mô tả giao diện người dùng.

C. Làm giảm số lượng bước cần thiết trong tài liệu.

\*D. Làm cho văn bản rõ ràng hơn về các hành động của người dùng và hệ thống.

**Question 13: Tại sao cần tạo ra các kịch bản "trời mưa" trong mỗi use case?**

A. Để giảm số lượng các yêu cầu chức năng cần xem xét.

\*B. Để dự đoán và xử lý các lỗi hoặc trường hợp ngoại lệ có thể xảy ra.

C. Để giúp hệ thống linh hoạt hơn trong việc thay đổi yêu cầu.

D. Để đáp ứng các yêu cầu về hiệu suất của hệ thống.

**Question 14: Bạn đang thực hiện đánh giá một use case, nhưng mô tả quá trừu tượng và khó hiểu. Bạn sẽ làm gì để cải thiện tài liệu?**

\*A. Cụ thể hóa các hành động của người dùng và phản hồi của hệ thống bằng cách sử dụng giọng chủ động.

B. Sử dụng các từ ngữ chuyên môn để làm tăng tính chuyên sâu của tài liệu.

C. Bỏ qua các phần trừu tượng và tập trung vào giao diện người dùng.

D. Chuyển đổi toàn bộ văn bản sang dạng bảng biểu.

**Question 15:Trong quá trình đánh giá, khách hàng phàn nàn rằng hệ thống không đáp ứng được yêu cầu của họ, mặc dù yêu cầu đã được phê duyệt trước đó. Bạn sẽ xử lý thế nào?**

A. Tạm dừng dự án và yêu cầu khách hàng cung cấp yêu cầu mới.

B. Chuyển yêu cầu đến nhóm phát triển mà không thay đổi gì.

\*C. Thực hiện lại buổi đánh giá yêu cầu với khách hàng để tìm ra sự khác biệt và cập nhật tài liệu.

D. Loại bỏ ý kiến của khách hàng vì không phù hợp với giai đoạn hiện tại.

**Question 16: Ai cần tham gia vào buổi đánh giá yêu cầu?**

\*A. Đại diện khách hàng, người dùng cuối, nhân viên marketing, và các bên liên quan khác.

B. Nhóm quản lý dự án.

C. Chỉ đội ngũ phát triển.

D. Chỉ các lập trình viên.

**Question 17: Tại sao cần xem xét các yêu cầu trước khi bắt đầu mã hóa?**

A. Để giảm chi phí phát triển.

B. Để tiết kiệm thời gian lập trình.

C. Để tránh các thay đổi yêu cầu trong tương lai.

\*D. Để xác minh rằng các yêu cầu không bị hiểu sai.

**Question 18: Một use case không có "kịch bản thay thế" thường cho thấy điều gì?**

A. Nó đã được phân tích đầy đủ.

B. Nó chỉ dành cho các hệ thống nhỏ.

C. Nó không cần thiết trong quy trình phát triển.

\*D. Nó thiếu các kịch bản xử lý lỗi hoặc trường hợp đặc biệt.

**Question 19: Tại sao việc mời đại diện khách hàng tham gia buổi đánh giá yêu cầu lại quan trọng?**

A. Để xác định chi phí thực hiện dự án.

B. Để tăng cường mối quan hệ giữa khách hàng và nhà phát triển.

C. \*Để đảm bảo rằng yêu cầu hệ thống phản ánh chính xác nhu cầu thực tế của khách hàng.\*

D. Để tiết kiệm thời gian hoàn thành dự án.

**Question 20: Khi mô tả các bước trong use case, tại sao cần liên kết các bước này với các đối tượng miền?**

\*A. Để đảm bảo rằng mô hình miền hỗ trợ trực tiếp các yêu cầu của hệ thống.

B. Để tăng tính trừu tượng của tài liệu.

C. Để tránh sử dụng các thuật ngữ không cần thiết.

D. Để giảm bớt số lượng các bước.

**Question 21:Tại sao việc viết use case theo ngôn ngữ chủ động lại quan trọng?**

A. Để tách biệt yêu cầu chức năng khỏi giao diện người dùng.

B. Để giúp lập trình viên hiểu rõ hơn về cách triển khai yêu cầu.

\*C. Để làm cho tài liệu trở nên dễ đọc hơn với khách hàng.

D. Để giảm thời gian kiểm thử hệ thống.

**Question 22: Tại sao việc đánh giá các use case cần có sự tham gia của cả người dùng cuối?**

A. Người dùng cuối chịu trách nhiệm tài trợ cho dự án.

B. Người dùng cuối có kinh nghiệm về quy trình phát triển.

C. Người dùng cuối thường có quyền quyết định về mô hình thiết kế.

\*D. Người dùng cuối có thể cung cấp phản hồi thực tế về cách hệ thống sẽ được sử dụng.

**Question 23: Làm thế nào việc truy vết (traceability) các yêu cầu đến các use case giúp ích cho dự án?**

A. Tăng độ chi tiết của mô hình miền.

B. Giảm nhu cầu kiểm tra hệ thống.

C. Giảm thời gian viết các use case.

\*D. Đảm bảo rằng mọi yêu cầu đều được thực hiện và kiểm tra.

**Question 24: Tại sao việc đặt tên rõ ràng cho các đối tượng trong mô hình miền lại quan trọng?**

A. Để giúp khách hàng dễ hiểu hơn về các yêu cầu.

B. Để đơn giản hóa tài liệu hóa hệ thống.

\*C. Để đảm bảo rằng các lập trình viên hiểu rõ vai trò của từng đối tượng.

D. Để tăng tốc độ hoàn thành mô hình.

**Question 25: Một use case mô tả rằng "Hệ thống phải cho phép người dùng tìm kiếm sản phẩm". Để cải thiện, bạn sẽ làm gì?**

A. Xóa use case này vì nó quá cơ bản.

\*B. Thay đổi câu mô tả thành: "Người dùng nhập từ khóa tìm kiếm, hệ thống hiển thị danh sách sản phẩm phù hợp."

C. Tăng cường mô tả bằng cách thêm các bước xử lý lỗi vào phần chính.

D. Liên kết use case với mô hình miền mà không thay đổi nội dung.

**Question 26: Mục đích của buổi xem xét yêu cầu (requirements review session) là gì?**

A. Để kiểm tra tính khả thi của các yêu cầu kỹ thuật.

B. Để thiết kế giao diện người dùng cho hệ thống.

\*C. Để đảm bảo rằng hệ thống được mô tả khớp với các yêu cầu.

D. Để đánh giá khả năng mở rộng của hệ thống.

**Question 27: Lợi ích của việc sử dụng ngôn ngữ chủ động trong mô tả use case là gì?**

A. Loại bỏ hoàn toàn các yêu cầu chức năng.

B. Giảm thời gian viết tài liệu.

\*C. Làm cho các yêu cầu rõ ràng hơn và dễ hiểu hơn.

D. Hạn chế số lượng lỗi khi mã hóa.

**Question 28 :Làm thế nào để tránh tình trạng "intermangled" trong đặc tả use case?**

\*A. Ghi lại các yêu cầu chức năng trong tài liệu riêng biệt.

B. Kết hợp các yêu cầu vào mô hình miền.

C. Sử dụng các thuật ngữ trừu tượng.

D. Loại bỏ hoàn toàn các yêu cầu chức năng.

**Question 29: Việc sử dụng giao diện người dùng (GUI) giúp cải thiện đặc tả use case như thế nào?**

A. Tăng khả năng tái sử dụng của các yêu cầu chức năng.

B. Loại bỏ nhu cầu sử dụng sơ đồ use case.

C. Giảm thời gian phát triển giao diện.

\*D. Làm rõ hành vi của người dùng và phản hồi của hệ thống.