# Giải đáp câu hỏi SWE (đã cập nhật)

Font: Times New Roman. Trình bày: đơn giản, chia phần rõ ràng.

## Phần 2.1: Câu hỏi trắc nghiệm (Đáp án)

**Câu 1:** B. Workflow lấy yêu cầu

**Câu 2:** B. Pha làm rõ (Elaboration)

**Câu 3:** C. Mức 4

**Câu 4:** B. Khởi đầu, làm rõ, xây dựng, chuyển giao

**Câu 5:** D. Workflow kiểm thử

**Câu 6:** C. Quy trình không ổn định, phụ thuộc vào cá nhân

**Câu 7:** B. Mô hình lặp và tăng trưởng (Iterative & Incremental)

**Câu 8:** A. Quy trình được cải tiến liên tục

**Câu 9:** C. Thiết kế kiến trúc và chi tiết hệ thống

**Câu 10:** B. Capability Maturity Model

## Phần 2.2: Câu hỏi trả lời ngắn (Giải thích ngắn gọn)

1. Pha khởi đầu trong tiến trình thống nhất là gì?

-> Trả lời: Pha khởi đầu (Inception) xác định mục tiêu dự án, phạm vi, các bên liên quan, các rủi ro lớn ban đầu và tạo cơ sở để quyết định có tiếp tục đầu tư hay không.

2. Mục tiêu của workflow lấy yêu cầu là gì?

-> Trả lời: Thu thập, phân tích và làm rõ yêu cầu của khách hàng, chuyển đổi nhu cầu thành các yêu cầu chức năng và phi chức năng có thể kiểm chứng được.

3. Tiến trình thống nhất gồm bao nhiêu pha chính?

-> Trả lời: Gồm 4 pha chính: Khởi đầu (Inception), Làm rõ (Elaboration), Xây dựng (Construction), Chuyển giao (Transition).

4. Sự khác nhau giữa CMM mức 2 và mức 3 là gì?

-> Trả lời: Mức 2 (Managed): tổ chức có quy trình cơ bản được quản lý; Mức 3 (Defined): quy trình được chuẩn hóa, định nghĩa và áp dụng rộng rãi, có tài liệu hướng dẫn và điều chỉnh trên toàn tổ chức.

5. Workflow kiểm thử có nhiệm vụ gì?

-> Trả lời: Thiết kế các ca kiểm thử, thực hiện kiểm thử để phát hiện lỗi, xác minh và xác nhận phần mềm đáp ứng yêu cầu; hỗ trợ chất lượng trong suốt vòng đời.

6. Mô hình CMM có bao nhiêu mức?

-> Trả lời: Mô hình CMM cổ điển có 5 mức: 1 (Initial), 2 (Managed), 3 (Defined), 4 (Quantitatively Managed), 5 (Optimizing).

7. Khác biệt giữa mô hình thác nước và mô hình lặp là gì?

-> Trả lời: Thác nước: các pha thực hiện tuần tự, ít cho phép thay đổi sau khi đã qua pha; Lặp (iterative): thực hiện theo vòng lặp nhỏ, cho phép phản hồi và sửa đổi trong các lần lặp, thích hợp với yêu cầu thay đổi.

8. Tiến trình thống nhất có phải là mô hình lặp không?

-> Trả lời: Có. Tiến trình thống nhất là mô hình lặp và gia tăng (iterative and incremental): dự án được triển khai qua nhiều vòng lặp, mỗi vòng cải thiện và bổ sung chức năng.

9. Mục đích của workflow thiết kế là gì?

-> Trả lời: Xác định kiến trúc hệ thống, thiết kế các thành phần và giao diện, chuyển yêu cầu thành thiết kế chi tiết để phục vụ việc cài đặt.

10. CMM mức 5 tập trung vào điều gì?

-> Trả lời: Tập trung vào cải tiến quy trình liên tục thông qua phân tích số liệu, phương pháp chủ động để tối ưu năng lực và hiệu quả của quy trình.

## Phần 2.3: Câu hỏi thảo luận nhóm (Gợi ý trả lời / điểm thảo luận)

1. Thảo luận về vai trò của từng workflow trong tiến trình phát triển phần mềm.

- Workflow lấy yêu cầu: xác định mong muốn khách hàng.

- Workflow phân tích & thiết kế: chuyển thành giải pháp kỹ thuật.

- Workflow cài đặt: hiện thực hóa chức năng.

- Workflow kiểm thử: đảm bảo chất lượng.

- Workflow quản lý cấu hình/ dự án/ môi trường: hỗ trợ điều phối, thay đổi và triển khai.

2. Phân biệt mô hình vòng đời thác nước và tiến trình thống nhất.

- Thác nước: tuần tự, dễ quản lý nhưng kém linh hoạt với thay đổi.

- Tiến trình thống nhất: lặp, quản lý rủi ro tốt hơn, thích hợp dự án lớn, phức tạp.

3. Thảo luận về các ưu và nhược điểm của mô hình lặp và tăng trưởng.

- Ưu: giảm rủi ro, nhận phản hồi sớm, thích nghi thay đổi.

- Nhược: yêu cầu quản lý lặp kỹ, có thể tăng chi phí nếu lặp không kiểm soát.

4. Vì sao mô hình CMM được sử dụng rộng rãi trong quản lý chất lượng phần mềm?

- CMM cung cấp khung đánh giá mức trưởng thành của quy trình, giúp cải tiến có hệ thống, giảm lỗi, tăng tin cậy và hiệu quả.

5. Thảo luận về các khó khăn khi áp dụng mô hình CMM trong thực tế.

- Chi phí và thời gian để triển khai quy trình chuẩn hóa.

- Kháng cự thay đổi từ nhân sự.

- Yêu cầu dữ liệu định lượng để đánh giá mức cao hơn.

6. Đề xuất các giải pháp để cải tiến quy trình phát triển phần mềm.

- Áp dụng công cụ tự động hóa (CI/CD, kiểm thử tự động).

- Đào tạo nhân sự, xây dựng template và checklist.

- Thu thập dữ liệu và cải tiến theo chu kỳ.

7. Phân tích ưu điểm của việc áp dụng tiến trình thống nhất trong các dự án lớn.

- Quản lý rủi ro tốt hơn, kiểm soát kiến trúc, phù hợp phân chia công việc theo giai đoạn và lặp.

8. Thảo luận về sự cần thiết của việc kiểm thử trong từng pha của tiến trình thống nhất.

- Kiểm thử sớm phát hiện lỗi, giảm chi phí sửa sau này; kiểm thử đơn vị, tích hợp và hệ thống nên xen kẽ với phát triển.

9. So sánh giữa mô hình CMM mức 4 và mức 5.

- Mức 4: quản lý định lượng (sử dụng số liệu để kiểm soát hiệu suất).

- Mức 5: tối ưu hóa liên tục (cải tiến dựa trên phân tích nguyên nhân và đổi mới).

10. Đề xuất cách tổ chức hoạt động nhóm trong workflow lấy yêu cầu.

- Phân vai: analyst, product owner, stakeholder liaison.

- Sử dụng workshops, user story mapping, prototyping để làm rõ yêu cầu.

- Thiết lập tiêu chí chấp nhận rõ ràng.

## Phần 2.4: Câu hỏi tình huống (Đề xuất giải pháp cụ thể)

1. Một công ty phát triển phần mềm gặp khó khăn khi yêu cầu của khách hàng liên tục thay đổi trong pha xây dựng. Đội phát triển nên làm gì để giải quyết vấn đề này?

- Thiết lập quy trình quản lý thay đổi (change control).

- Ưu tiên các yêu cầu theo giá trị và rủi ro; áp dụng iteration ngắn hơn hoặc chuyển sang mô hình Agile nếu phù hợp.

- Thương thảo và cập nhật hợp đồng/điều kiện chấp nhận; dùng prototyping để làm rõ yêu cầu.

2. Trong pha chuyển giao của tiến trình thống nhất, khách hàng yêu cầu bổ sung thêm tính năng mới. Đội phát triển nên xử lý ra sao?

- Đánh giá tác động (chi phí, thời gian, rủi ro).

- Nếu là tính năng nhỏ có thể đưa vào bản vá/phiên bản tiếp theo; nếu quan trọng và cần ngay, thiết lập scope change, lên kế hoạch bổ sung và test.

3. Dự án phát triển phần mềm bị trễ tiến độ do lỗi phát sinh liên tục trong quá trình kiểm thử. Là trưởng dự án, bạn sẽ làm gì?

- Phân tích nguyên nhân gốc (root cause) của lỗi: yêu cầu mơ hồ, code kém, thiếu test case.

- Tăng cường review, kiểm thử tự động, pair programming, và cải thiện quy trình mã hóa (coding standards).

- Điều chỉnh kế hoạch, tăng tài nguyên tạm thời hoặc giảm scope để bảo đảm tiến độ giao hàng.

4. Trong workflow thiết kế, kiến trúc sư phần mềm muốn thay đổi thiết kế ban đầu để cải thiện hiệu suất. Đội phát triển nên xử lý thế nào?

- Yêu cầu kiến trúc sư trình bày phân tích lợi ích/chi phí, benchmark và rủi ro.

- Thực hiện prototype nhỏ hoặc proof-of-concept trước khi áp dụng rộng rãi.

- Nếu chấp nhận, cập nhật thiết kế, thông báo thay đổi tới nhóm và điều chỉnh test plan.

5. Khách hàng yêu cầu rút ngắn thời gian phát triển dự án mà không thay đổi yêu cầu. Đội phát triển nên phản ứng ra sao?

- Đàm phán lại phạm vi, ưu tiên các tính năng thiết yếu (MVP).

- Tăng nguồn lực hoặc áp dụng kỹ thuật tăng tốc (parallel tasks, reuse).

- Nêu rõ rủi ro về chất lượng và đặt mốc kiểm thử/triển khai phù hợp.

6. Một công ty nhỏ muốn áp dụng mô hình CMM nhưng gặp khó khăn do thiếu nguồn lực. Hãy đề xuất giải pháp.

- Áp dụng từng bước: bắt đầu từ các thực hành cơ bản (CMM level 2) như quản lý yêu cầu và kiểm soát phiên bản.

- Sử dụng công cụ miễn phí/chi phí thấp và đào tạo nội bộ; tối ưu hóa quy trình nhẹ (lightweight processes).

7. Trong workflow lấy yêu cầu, khách hàng cung cấp thông tin không rõ ràng. Đội phát triển cần làm gì?

- Tổ chức workshop/meeting trực tiếp để làm rõ, dùng kỹ thuật như user story, acceptance criteria, mockup/prototype.

- Ghi lại biên bản, yêu cầu xác nhận bằng văn bản trước khi tiến hành phát triển.

8. Một dự án gặp rủi ro cao trong pha khởi đầu do thiếu tài liệu yêu cầu rõ ràng. Đội phát triển nên làm gì?

- Tập trung vào việc hoàn thiện tài liệu yêu cầu tối thiểu cần thiết để giảm rủi ro (use cases, non-functional requirements).

- Áp dụng kỹ thuật giảm rủi ro: spike, prototype, proof-of-concept để kiểm chứng giả thiết.

9. Dự án phần mềm lớn có nhiều nhóm phát triển ở các địa điểm khác nhau. Làm thế nào để đảm bảo các nhóm phối hợp hiệu quả?

- Thiết lập quy trình giao tiếp rõ ràng (daily/weekly sync), dùng công cụ quản lý công việc chung (issue tracker, CI), định nghĩa API/contract rõ ràng.

- Chuẩn hóa coding conventions, test contract và có các integration sprint để hoà hợp.

10. Một công ty phát triển phần mềm gặp khó khăn trong việc quản lý quy trình do không có chuẩn hóa. Hãy đề xuất giải pháp.

- Bắt đầu xây dựng các quy trình chuẩn hóa cốt lõi (yêu cầu, code review, test, release).

- Triển khai template, checklist, và đào tạo; áp dụng công cụ tự động để bắt buộc quy trình (CI checks).