

TRẢ LỜI CÂU HỎI THẢO LUẬN NHÓM

MSSV: N22DCAT050

Họ và tên: Trần Gia Thân

Lớp: D22CQAT01-N

Nhóm: 03

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

Workflow là chuỗi các bước hoặc quy trình cần thiết để phát triển phần mềm. Mỗi workflow có một vai trò quan trọng:

- Yêu cầu (Requirement Workflow): Xác định và thu thập yêu cầu từ khách hàng.
- Phân tích (Analysis Workflow): Hiểu rõ yêu cầu, xác định các giải pháp khả thi.
- Thiết kế (Design Workflow): Tạo ra kiến trúc và thiết kế chi tiết cho phần mềm.
- Triển khai (Implementation Workflow): Viết mã dựa trên thiết kế.
- Kiểm thử (Testing Workflow): Đảm bảo phần mềm hoạt động đúng yêu cầu, không có lỗi.
- Triển khai và bảo trì (Deployment & Maintenance Workflow): Đưa sản phẩm ra thị trường và cập nhật khi cần.

Vai trò của từng workflow giúp phần mềm được phát triển có kế hoạch, hạn chế lỗi và đáp ứng nhu cầu khách hàng.

Ví dụ: Phát triển ứng dụng đặt đồ ăn trực tuyến.

- Yêu cầu: Khách hàng yêu cầu chức năng đặt món, thanh toán, theo dõi đơn hàng.
- Phân tích: Xác định các đối tượng như người dùng, nhà hàng, shipper; tìm hiểu các rủi ro.
- Thiết kế: Vẽ sơ đồ kiến trúc, thiết kế giao diện người dùng (UI).
- Triển khai: Lập trình chức năng đặt món, thanh toán.
- Kiểm thử: Thử nghiệm xem người dùng có thể đặt món và thanh toán thành công không.
- Triển khai và bảo trì: Đưa ứng dụng lên App Store, sửa lỗi nếu người dùng phản hồi.

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

- Mô hình vòng đời thác nước (Waterfall):

Tiến trình tuyến tính, từng bước: Yêu cầu → Thiết kế → Triển khai → Kiểm thử → Bảo trì.

Ưu điểm: Đơn giản, dễ quản lý.

Nhược điểm: Không linh hoạt, khó thay đổi khi đã sang giai đoạn sau.

- Tiến trình thống nhất (RUP - Rational Unified Process):

Chia thành 4 pha: Khởi đầu → Lập kế hoạch → Xây dựng → Chuyển giao.

Có các workflow lặp lại trong từng pha.

Ưu điểm: Linh hoạt, thích ứng với thay đổi.

Nhược điểm: Phức tạp, cần tài liệu nhiều.

Waterfall phù hợp dự án nhỏ, yêu cầu rõ ràng. RUP phù hợp dự án lớn, yêu cầu thay đổi liên tục.

Ví dụ:

- Thác nước: Phát triển phần mềm quản lý thư viện. Tất cả yêu cầu được thu thập xong mới thiết kế, lập trình và kiểm thử. Nếu phát hiện thiếu sót, phải quay lại bước đầu.
- Tiến trình thống nhất (RUP): Phát triển game. Chia thành các phiên bản nhỏ (v1: chơi đơn, v2: chơi online). Mỗi phiên bản đều trải qua các workflow lặp lại.

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

- Mô hình lặp (Iterative):

Xây dựng phần mềm qua các vòng lặp nhỏ, cải tiến dần.

Ưu điểm: Phát hiện lỗi sớm, linh hoạt.

Nhược điểm: Quản lý phức tạp, tốn thời gian.

- Mô hình tăng trưởng (Incremental):

Xây dựng phần mềm theo từng phần nhỏ, tích hợp dần.

Ưu điểm: Người dùng có thể sử dụng sớm một phần sản phẩm.

Nhược điểm: Thiết kế tổng thể ban đầu có thể chưa tối ưu.

Cả hai mô hình đều giúp giảm rủi ro so với Waterfall.

Ví dụ: Xây dựng website bán hàng.

- Mô hình lặp:

Vòng lặp 1: Xây dựng chức năng xem sản phẩm.

Vòng lặp 2: Thêm chức năng giỏ hàng.

Vòng lặp 3: Thêm thanh toán trực tuyến.

- Ưu điểm: Người dùng có thể góp ý sau mỗi vòng.
- Nhược điểm: Quản lý nhiều phiên bản phức tạp.
- Mô hình tăng trưởng:

Phiên bản 1: Cho phép xem sản phẩm.

Phiên bản 2: Thêm giỏ hàng.

Phiên bản 3: Thêm đánh giá sản phẩm.

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

CMM (Capability Maturity Model) giúp đo lường và cải thiện quy trình phát triển phần mềm qua 5 mức độ trưởng thành.

- Lý do sử dụng rộng rãi:

Cải thiện chất lượng và hiệu suất.

Giảm rủi ro và chi phí.

Nâng cao sự hài lòng của khách hàng.

Giúp tổ chức có quy trình rõ ràng và chuẩn hóa.

Ví dụ: Công ty ABC muốn đạt CMM cấp 3 để ký hợp đồng lớn.

- Khi áp dụng CMM:

- Quy trình phát triển phần mềm chuẩn hóa.
- Chất lượng sản phẩm cải thiện, giảm lỗi.
- Được khách hàng tin tưởng hơn.

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

- Khó khăn phổ biến:

Tốn kém chi phí và thời gian.

Cần thay đổi văn hóa và thói quen làm việc.

Quản lý tài liệu phức tạp.

Thiếu sự hỗ trợ từ lãnh đạo hoặc nhân viên không hợp tác.

Cần sự cam kết từ mọi cấp trong tổ chức để áp dụng thành công.

Ví dụ: Công ty nhỏ áp dụng CMM gặp khó khăn:

- Nhân viên không quen với việc viết nhiều tài liệu.
- Chi phí đào tạo cao.
- Lãnh đạo chưa thấy rõ lợi ích ngắn hạn nên thiếu hỗ trợ.

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

- Giải pháp khả thi:

Áp dụng phương pháp Agile để linh hoạt hơn.

Tăng cường giao tiếp giữa các bên liên quan.
Sử dụng công cụ hỗ trợ quản lý dự án.
Thường xuyên kiểm thử và đánh giá sản phẩm.
Đào tạo và nâng cao kỹ năng cho đội ngũ.

Ví dụ: Đội phát triển phần mềm bị chậm tiến độ.

Giải pháp:

- Áp dụng Scrum để chia nhỏ công việc theo sprint 2 tuần.
- Sử dụng công cụ như Jira để quản lý tiến độ.
- Thường xuyên họp daily meeting để cập nhật tình hình.

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

- Ưu điểm nổi bật:
 - Quản lý rủi ro tốt hơn.
 - Linh hoạt khi yêu cầu thay đổi.
 - Phân chia công việc rõ ràng giữa các team.
 - Có thể kiểm thử và đánh giá sớm.
 - Đáp ứng tốt cho dự án phức tạp, nhiều bên liên quan.

Ví dụ: Phát triển hệ thống quản lý bệnh viện.

Ưu điểm:

- Chia thành các giai đoạn nhỏ (quản lý bệnh nhân, quản lý thuốc, lịch hẹn).
- Có thể triển khai từng phần, không cần chờ toàn bộ hoàn thành.
- Đội ngũ có thể phát hiện và sửa lỗi sớm.

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

- Lý do cần kiểm thử liên tục:
 - Phát hiện và sửa lỗi sớm, giảm chi phí về sau.
 - Đảm bảo yêu cầu được đáp ứng đúng.
 - Giúp cải thiện chất lượng tổng thể.
 - Tránh tích lũy lỗi lớn khi đến giai đoạn cuối.

Kiểm thử sớm và thường xuyên giúp sản phẩm ổn định và đáng tin cậy.

Ví dụ: Khi phát triển app ngân hàng, nếu không kiểm thử sớm:

- Chức năng chuyển tiền bị lỗi, gây mất uy tín.

- Sửa lỗi sau khi phát hành tốn kém hơn.

Kiểm thử từng pha giúp:

- Phát hiện lỗi logic khi mới thiết kế.
- Đảm bảo các chức năng an toàn, tránh sự cố.

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

Tiêu chí	CMM Mức 4 (Được quản lý)	CMM Mức 5 (Tối ưu hóa)
Mục tiêu	Đo lường, kiểm soát quy trình	Liên tục cải tiến quy trình
Đặc điểm	Quy trình ổn định, có dữ liệu đo lường	Tập trung đổi mới và ngăn ngừa lỗi
Lợi ích	Hiệu suất được kiểm soát tốt	Cải thiện hiệu suất liên tục

Ví dụ:

- Công ty X (CMM mức 4): Có dữ liệu đo lường chất lượng phần mềm, biết dự án thường trễ bao lâu.
- Công ty Y (CMM mức 5): Không chỉ đo lường mà còn tự động cải thiện quy trình để giảm trễ.

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

- Cách tổ chức hiệu quả:

Bước 1: Xác định người đại diện khách hàng.

Bước 2: Tổ chức cuộc họp nhóm (Brainstorming, phỏng vấn, khảo sát).

Bước 3: Phân chia vai trò (người ghi chép, người hỏi, người quan sát).

Bước 4: Tổng hợp yêu cầu, xác minh với khách hàng.

Bước 5: Lập tài liệu và chia sẻ với toàn bộ nhóm.

Đảm bảo giao tiếp rõ ràng và khách hàng tham gia tích cực.

Ví dụ: Phát triển app học online.

Cách tổ chức:

- Bước 1: Họp với khách hàng (giáo viên, học sinh).
- Bước 2: Phân công:
 - Người A: Ghi chép yêu cầu.
 - Người B: Đặt câu hỏi làm rõ.
 - Người C: Quan sát phản ứng khách hàng.

- Bước 3: Tổng hợp yêu cầu thành tài liệu.
- Bước 4: Gửi khách hàng xác nhận trước khi thiết kế.