

TRẢ LỜI CÂU HỎI TÌNH HUỐNG

MSSV: N22DCAT050

Họ và tên: Trần Gia Thân

Lớp: D22CQAT01-N

Nhóm: 03

1. Khách hàng liên tục thay đổi yêu cầu trong pha xây dựng. Đội phát triển nên làm gì?

Vấn đề: Khi đang xây dựng phần mềm, khách hàng liên tục thêm, bớt hoặc sửa yêu cầu. Điều này dễ gây trễ tiến độ, tốn chi phí và làm giảm chất lượng.

Giải pháp chi tiết:

- Áp dụng quy trình quản lý thay đổi:
 - Thiết lập mẫu yêu cầu thay đổi để khách hàng điền.
 - Đánh giá tác động (về thời gian, chi phí, nhân lực) cho từng thay đổi.
- Thảo luận với khách hàng:
 - Giải thích ảnh hưởng của việc thay đổi liên tục.
 - Thống nhất phiên bản yêu cầu chốt trước khi xây dựng.
- Áp dụng mô hình phát triển linh hoạt (Agile):
 - Từng sprint sẽ có danh sách yêu cầu cố định, hạn chế thay đổi giữa sprint.

Ví dụ: Đang phát triển ứng dụng đặt vé máy bay, khách muốn thêm "tính năng chat hỗ trợ" sau khi xây dựng xong phần đặt vé. Đội phát triển thảo luận và đưa tính năng này vào phiên bản cập nhật sau thay vì thêm ngay.

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

Vấn đề: Sau khi phát triển xong, khách đột ngột muốn thêm tính năng mới (ví dụ: thêm đăng nhập bằng Facebook).

Giải pháp chi tiết:

Đánh giá tính cần thiết: Tính năng này có thật sự cấp thiết không? Có thể để cho phiên bản sau không?

Thảo luận và lập hợp đồng bổ sung: Nếu cần gấp, hai bên sẽ ký thêm hợp đồng với chi phí và thời gian bổ sung.

Phân bổ lại nguồn lực: Đảm bảo đội ngũ có đủ nhân lực để phát triển thêm mà không ảnh hưởng chất lượng.

Ví dụ: Khi giao website bán hàng, khách muốn thêm "đánh giá sản phẩm". Đội phát triển thỏa thuận bổ sung tính năng này trong bản cập nhật v1.1 sau khi đưa sản phẩm ra thị trường.

3. Dự án bị trễ tiến độ do lỗi phát sinh khi kiểm thử. Là trưởng dự án, bạn làm gì?

Vấn đề: Lỗi liên tục xuất hiện khi kiểm thử, dẫn đến trì hoãn.

Giải pháp chi tiết:

- Xác định nguyên nhân:
 - Lỗi do hiểu sai yêu cầu?
 - Thiếu kiểm thử đơn vị (unit test)?
 - Do vội vàng code?
- Hành động khắc phục:
 - Bổ sung kiểm thử tự động.
 - Áp dụng code review nghiêm ngặt.
 - Chia nhỏ module để kiểm thử riêng.
- Điều chỉnh tiến độ: Đàm phán với khách về thời gian giao hàng mới hợp lý.

🔗 Ví dụ: Phát triển phần mềm quản lý kho, phát hiện lỗi khi nhập hàng. Đội phát triển đã thêm unit test cho từng chức năng và tổ chức kiểm thử chéo để giảm lỗi.

4. Trong thiết kế, kiến trúc sư muốn thay đổi thiết kế ban đầu. Đội nên xử lý thế nào?

Vấn đề: Giữa dự án, kiến trúc sư muốn chuyển từ MySQL sang MongoDB để tăng hiệu suất.

Giải pháp chi tiết:

- Phân tích tác động: Thay đổi này ảnh hưởng gì đến code hiện tại? Chi phí và thời gian tăng bao nhiêu?
- Thảo luận với khách hàng: Nếu thay đổi có lợi lâu dài và khách đồng ý, hãy cập nhật thiết kế.
- Cập nhật tài liệu: Sửa tài liệu thiết kế, thông báo đến các nhóm liên quan.

Ví dụ: Đang dùng cơ sở dữ liệu quan hệ, phát hiện tốc độ truy vấn chậm. Sau khi phân tích, quyết định chuyển sang NoSQL và gia hạn dự án thêm 2 tuần để triển khai.

5. Khách yêu cầu rút ngắn thời gian mà không thay đổi yêu cầu. Đội phát triển nên làm gì?

Vấn đề: Dự án 2 tháng bị yêu cầu rút xuống 1 tháng.

Giải pháp chi tiết:

- Đàm phán lại: Giải thích rủi ro khi gấp rút tiến độ.
- Tăng nguồn lực: Thêm nhân viên hoặc thuê freelancer hỗ trợ.
- Ưu tiên chức năng chính: Phát hành phiên bản MVP trước.
- Phát triển song song: Frontend, backend và kiểm thử tiến hành cùng lúc.

Ví dụ: Làm app đặt đồ ăn, ưu tiên chức năng đặt món ra trước, còn tính năng "đánh giá" để sau.

6. Công ty nhỏ muốn áp dụng mô hình CMM nhưng thiếu nguồn lực. Giải pháp?

Vấn đề: Công ty startup muốn nâng cao quy trình nhưng thiếu người.

Giải pháp chi tiết:

- Áp dụng dần dần: Bắt đầu từ CMM cấp 2 (quản lý yêu cầu, kiểm thử).
- Tận dụng công cụ miễn phí: Sử dụng GitHub, Trello cho quản lý.
- Thuê tư vấn ngắn hạn: Giúp công ty xây dựng quy trình nhanh hơn.

Ví dụ: Sử dụng Jira để quản lý task và Google Docs để lưu trữ yêu cầu.

7. Trong lấy yêu cầu, khách hàng cung cấp thông tin không rõ. Đội nên làm gì?

Vấn đề: Khách nói "muốn giao diện dễ dùng" nhưng không mô tả chi tiết.

Giải pháp chi tiết:

- Phỏng vấn sâu: Hỏi thêm "đối tượng người dùng là ai?", "có mẫu nào tham khảo không?".
- Dùng wireframe: Tạo bản vẽ giao diện sơ bộ cho khách xem.
- Tổ chức workshop: Mời khách tham gia để hiểu rõ hơn.

Ví dụ: Đưa ra 2 mẫu giao diện (đơn giản & hiện đại) cho khách chọn.

8. Dự án gặp rủi ro cao do thiếu tài liệu yêu cầu rõ ràng. Đội nên làm gì?

Vấn đề: Không có tài liệu chi tiết, dễ làm sai yêu cầu.

Giải pháp chi tiết:

- Yêu cầu khách bổ sung: Cung cấp mô tả chi tiết hoặc video mô phỏng.
- Sử dụng use case: Mô tả tình huống sử dụng để hiểu rõ hơn.
- Làm bản mẫu: Tạo prototype để khách xác nhận.

Ví dụ: Với app ngân hàng, làm bản demo chức năng đăng nhập để khách xem.

9. Dự án lớn có nhiều nhóm ở các địa điểm khác nhau. Làm sao phối hợp hiệu quả?

Vấn đề: Nhóm ở Việt Nam, Mỹ, Ấn Độ dễ gặp chênh lệch giờ, giao tiếp kém.

Giải pháp chi tiết:

- Họp định kỳ: Dùng Zoom, Teams mỗi tuần.
- Công cụ quản lý: Dùng Jira/Trello để theo dõi tiến độ.
- Giao tiếp rõ ràng: Tất cả tài liệu để trên Google Drive, ghi chú đầy đủ.
- Thời gian chồng lấp: Chọn giờ họp phù hợp cho tất cả (ví dụ 9h sáng Việt Nam, 10h tối Mỹ).

Ví dụ: Slack cho chat nhanh, Confluence cho lưu tài liệu.

10. Công ty gặp khó khi quản lý quy trình do không có chuẩn hóa. Giải pháp?

Vấn đề: Mỗi dự án làm một kiểu, không nhất quán.

Giải pháp chi tiết:

- Xây dựng quy trình chuẩn:
 - Quy trình lấy yêu cầu, kiểm thử, triển khai.
- Đào tạo nhân viên: Hướng dẫn mọi người tuân thủ quy trình.
- Công cụ hỗ trợ: Dùng Git, Jira để theo dõi tiến độ và lỗi.
- Rút kinh nghiệm: Sau mỗi dự án tổ chức họp lessons learned.

Ví dụ: Dùng GitFlow cho quản lý code, Trello cho quản lý task.