BÀI BÁO CÁO 15

1. Nếu xung đột với Developer thì cần làm gì?

Khi có xung đột với một lập trình viên hoặc nhóm phát triển, quá trình giải quyết xung đột có thể được thực hiện bằng cách sau:

Thảo luận và lắng nghe: Đầu tiên, quan trọng hơn hết, bạn nên bắt đầu bằng việc thảo luận và lắng nghe. Làm việc chung để hiểu các góc nhìn và quan điểm của nhau. Hãy thử giải quyết vấn đề qua cuộc trò chuyện mà không phải qua email hoặc thông điệp viết.

Xác định vấn đề cụ thể: Hãy đảm bảo rằng bạn hiểu rõ vấn đề cụ thể đang gây xung đột. Điều này giúp tập trung vào giải pháp.

Tìm kiếm giải pháp: Hãy tìm kiếm cách để giải quyết vấn đề cụ thể mà không phải tìm lỗi hoặc chỉ trích. Hãy đề xuất giải pháp hoặc thảo luận về những cách để cải thiện tình hình.

Thỏa thuận: Cố gắng đạt được thỏa thuận hoặc giải quyết thỏa đáng. Có thể cần sự hy sinh từ cả hai bên để đảm bảo rằng dự án tiến triển một cách trơn tru.

Sử dụng quản lý xung đột (nếu cần): Trong trường hợp xung đột không thể giải quyết một cách hòa bình, có thể cần sự can thiệp của người quản lý hoặc người quản lý dự án để đưa ra quyết định.

Bảo lưu bằng văn bản: Sau khi có thỏa thuận hoặc giải quyết xung đột, nên thể hiện nó bằng văn bản hoặc email. Điều này giúp tránh hiểu lầm trong tương lai.

Học hỏi từ kinh nghiệm: Xem xét xem xung đột có thể được xem là cơ hội để học hỏi và cải thiện quá trình làm việc chung. Hãy nắm bài học từ mỗi xung đột để tránh lặp lại chúng trong tương lai.

1. Lỗi được chấp nhận bởi ai?

Lỗi trong phát triển phần mềm có thể được chấp nhận bởi một số người hoặc bộ phận khác nhau trong một tổ chức hoặc dự án phụ thuộc vào tính nghiêm trọng và tầm ảnh hưởng của lỗi. Dưới đây là một số cách thức phân loại:

Lỗi cơ bản được phát hiện bởi lập trình viên (Developer): Những lỗi cơ bản hoặc khó nhận thấy trong mã nguồn ban đầu thường được lập trình viên phát hiện và tự sửa trước khi nó đi vào quá trình kiểm thử.

Lỗi phát hiện bởi người kiểm thử (Tester): Một số lỗi có thể thoát khỏi sự chú ý của lập trình viên và cần được kiểm tra bởi người kiểm thử. Nếu lỗi này không ảnh hưởng nhiều đến chức năng chính hoặc an ninh, người kiểm thử có thể chấp nhận nó và báo cáo cho nhóm phát triển để sửa.

Lỗi được chấp nhận bởi quản lý chất lượng (QA Manager): Các lỗi nhỏ hoặc không ảnh hưởng đến chức năng chính của sản phẩm có thể được quản lý chất lượng chấp nhận. Họ đánh giá tính nghiêm trọng và tầm ảnh hưởng của lỗi và quyết định xem liệu nó nên được sửa ngay lập tức hay có thể đợi cho bản cập nhật sau này.

Lỗi được chấp nhận bởi khách hàng hoặc người dùng cuối: Trong một số trường hợp, các lỗi có thể được chấp nhận bởi khách hàng hoặc người dùng cuối nếu chúng không ảnh hưởng đến khả năng sử dụng sản phẩm một cách cơ bản và không làm ảnh hưởng đến tính bảo mật hay thông tin cá nhân.

Lỗi cố ý (trong trường hợp cần thiết): Một số lỗi cố ý hoặc hạn chế được chấp nhận trong các tình huống nhất định, chẳng hạn như trong môi trường phát triển nhanh, để giúp tối ưu hóa thời gian và nguồn lực. Tuy nhiên, việc này cần phải thảo luận và đưa ra quyết định chính xác.

1. Khi nào lỗi được đóng?

Lỗi trong phát triển phần mềm thường được đóng khi chúng đã được xử lý và không còn tác động tiêu cực đến sản phẩm hoặc khi mục tiêu đã được đạt. Dưới đây là một số tình huống thường thấy khi lỗi có thể được đóng:

Lỗi đã được sửa: Sau khi lỗi đã được phát hiện và báo cáo, nó được giao cho lập trình viên để sửa. Sau khi lập trình viên đã sửa xong, lỗi được kiểm thử lại để đảm bảo rằng nó đã được sửa thành công. Nếu lỗi đã được sửa, nó có thể được đóng.

Lỗi không thể tái hiện: Đôi khi, lỗi không thể tái hiện hoặc mô phỏng sau khi đã thực hiện các bước nhất định. Nếu lỗi không thể tái hiện, nó có thể được đóng vì không thể kiểm tra hoặc xác minh thêm.

Lỗi không ảnh hưởng đến tính năng quan trọng: Nếu lỗi không ảnh hưởng nhiều đến tính năng chính của sản phẩm hoặc không gây ra các vấn đề quan trọng, nó có thể được đóng.

Lỗi không quan trọng: Các lỗi không quan trọng hoặc nhỏ không ảnh hưởng nhiều đến trải nghiệm người dùng cuối có thể được đóng nếu chúng không cần ưu tiên hoặc tài nguyên để sửa.

Lỗi là một phần của tính năng không hoàn thiện: Trong một số trường hợp, lỗi có thể xuất hiện trong giai đoạn phát triển sản phẩm và có thể không được xử lý ngay lập tức nếu chúng liên quan đến các tính năng chưa hoàn thiện hoặc chưa được phát hành.

Lỗi được đóng do quyết định cố ý: Trong một số tình huống, lỗi có thể được đóng cố ý nếu quyết định là chấp nhận nó để tối ưu hóa thời gian và tài nguyên.

Lỗi làm chết hệ thống, không thể thực hiện bất kỳ thao tác gì với hệ thống thuộc mức độ lỗi nào?

Mức độ của một lỗi trong phát triển phần mềm thường được đánh giá dựa trên sự nghiêm trọng và tác động của lỗi đó đối với hệ thống. Lỗi mà bạn mô tả là một lỗi nghiêm trọng, thường được gọi là "lỗi gây ra sự cố" hoặc "lỗi làm chết hệ thống," và nó ảnh hưởng nghiêm trọng đến hoạt động của hệ thống. Đây là một số đặc điểm của mức độ lỗi nghiêm trọng này:

Lỗi gây ra sự cố: Lỗi này có thể dẫn đến sự cố nghiêm trọng trong hệ thống. Hệ thống có thể không hoạt động hoặc bị tê liệt do lỗi này.

Không thể thực hiện bất kỳ thao tác gì: Người dùng hoặc quản trị viên không thể thực hiện bất kỳ thao tác nào trên hệ thống do tác động của lỗi. Hệ thống có thể bị đóng hoặc không đáp ứng đối với mọi yêu cầu.

Các lỗi cần giải quyết ngay lập tức: Lỗi này cần phải được ưu tiên và giải quyết ngay lập tức để khôi phục hoạt động của hệ thống.

Lỗi đe dọa tính toàn vẹn của dữ liệu: Mức độ lỗi này thường liên quan đến việc đe dọa đến tính toàn vẹn của dữ liệu. Dữ liệu có thể bị mất hoặc bị hỏng do tác động của lỗi.

Lỗi gây ra tổn thất lớn: Mức độ lỗi này thường gây ra tổn thất lớn về thời gian, tiền bạc và uy tín đối với tổ chức hoặc doanh nghiệp.

1. Một số hành vi không mong muốn của hệ thống, nhưng hệ thống vẫn hoạt động được thuộc mức độ lỗi nào?

Hành vi không mong muốn của hệ thống, nhưng hệ thống vẫn hoạt động được, thường thuộc vào mức độ lỗi "Minor" hoặc "Cosmetic" trong phạm vi bảo trì và phát triển phần mềm. Điều này có nghĩa là các lỗi này không gây ra sự cố nghiêm trọng và hệ thống vẫn có thể tiếp tục hoạt động, nhưng vẫn có những khuyết điểm hoặc hành vi không như mong đợi.

Dưới đây là một số ví dụ về hành vi không mong muốn thuộc vào mức độ lỗi "Minor" hoặc "Cosmetic":

Hiển thị văn bản không đúng: Ví dụ, một lỗi hiển thị văn bản trang web không đúng mà vẫn cho phép người dùng đọc thông tin.

Ảnh không hiển thị chính xác: Một lỗi có thể làm cho hình ảnh hoặc biểu đồ trông không đúng hoặc không đẹp mắt, nhưng hệ thống vẫn hoạt động bình thường.

Giao diện không tương thích với mọi trình duyệt: Một số trang web có thể hiển thị không đúng trên một số trình duyệt web cụ thể, nhưng vẫn hoạt động trên các trình duyệt khác.

Lỗi chính tả hoặc ngữ pháp không đáng kể: Lỗi chính tả hoặc ngữ pháp không đáng kể trong văn bản trang web có thể được coi là mức độ "Cosmetic."

Hiển thị cảnh báo không cần thiết: Một số hệ thống có thể hiển thị cảnh báo không cần thiết, nhưng chúng không ảnh hưởng đáng kể đến khả năng sử dụng.

1. Sau khi Developer thông báo lỗi được sửa thì Tester sẽ thực hiện giai đoạn nào?

Sau khi một lỗi được Developer thông báo đã sửa, Tester sẽ thực hiện giai đoạn kiểm thử sau đây:

Kiểm tra lỗi đã sửa (Retesting): Trong giai đoạn này, Tester kiểm tra xem lỗi đã được báo cáo trước đó đã được sửa chưa. Tester sẽ sử dụng bản vá hoặc phiên bản cập nhật mới để kiểm tra xem lỗi cụ thể đã biến mất chưa. Quá trình này đảm bảo rằng lỗi đã được sửa và không còn tồn tại nữa.

Kiểm tra liên quan (Regression Testing): Nếu lỗi đã được sửa bằng cách chỉnh sửa mã nguồn, Tester cũng cần thực hiện kiểm tra liên quan. Điều này đảm bảo rằng việc sửa lỗi không gây ra các lỗi khác ở các phần khác của phần mềm.

Báo cáo kết quả (Reporting): Sau khi kiểm tra lỗi đã sửa và kiểm tra liên quan, Tester sẽ báo cáo kết quả cho Developer và các thành viên khác trong dự án. Nếu lỗi đã được sửa thành công và không có lỗi liên quan, Tester sẽ thông báo rằng lỗi đã được đóng hoặc đã được giải quyết. Nếu lỗi vẫn tồn tại hoặc có sự thay đổi trong hành vi của phần mềm, Tester sẽ báo cáo chi tiết về tình trạng và lỗi còn lại.

Làm việc với Developer (Collaboration): Tester và Developer có thể phải làm việc cùng nhau để đảm bảo rằng lỗi đã được sửa đúng cách và không gây ra các vấn đề khác. Họ có thể cần thảo luận về cách sửa lỗi, đồng thuận về việc kiểm tra và kiểm tra lại, và đảm bảo rằng tất cả các vấn đề được giải quyết một cách hiệu quả.

1. Một báo cáo lỗi cần có những nội dung gì?

Bug number/id: giúp đội phát triển có thể dễ dàng kiểm tra lỗi được fix hay chưa

Bug title : gợi ý đủ những yêu cầu lỗi xãy ra

Priority: giúp cho việc đánh giá và ưu tiên để tập sửa chữa lỗi nghiêm trọng

Description: mô tả giúp đội phát triển hiểu được lỗi

Step: các bước thực hiện để kiểm tra lỗi

Enviroment: khi thực hiện báo cáo lỗi nên đưa ra nên thực hiện kiểm thử trên môi trường nào