

Báo cáo về 2 chủ đề The Broken Window Theory & The Boy Scout Rule

I. The Broken Window Theory

1. Giới thiệu

Lý thuyết The Broken Window Theory (cửa sổ vỡ) được James Q. Wilson và George L. Kelling giới thiệu vào năm 1982. Lý thuyết này cho rằng các dấu hiệu tội phạm, sai trái, có dấu hiệu lộn xộn có thể nhìn thấy được và không được sửa chữa kịp thời, sẽ tạo ra môi trường khuyến khích các phạm tội, dẫn đến các hành vi sai trái ngày càng nghiêm trọng hơn.

2. Nội dung lý thuyết

- Các nhà tâm lý học đều đồng ý rằng nếu một cửa sổ trong tòa nhà bị đập vỡ và không ai sửa chữa thì tất cả các cửa sổ còn cũng sẽ bị vỡ theo.
- Việc này xảy ra do mọi người cho rằng không ai quan tâm đến việc duy trì trật tự. Điều này còn có thể khuyến khích thêm những hành vi tiêu cực khác như vẽ bậy, xả rác hoặc thậm chí là tội phạm.
- Sự suy thoái của môi trường sẽ làm gia tăng tỷ lệ vi phạm luật lệ và tạo ra một vòng luân quần tiêu cực. Ngược lại, nếu các vấn đề nhỏ được xử lý ngay lập tức, nó sẽ gửi thông điệp rằng mọi người quan tâm và tôn trọng môi trường sống.

3. Ứng dụng lý thuyết trong thực tế

- Quản lý đô thị: Áp dụng các chính sách để xử lý những vi phạm nhỏ như vẽ bậy, đi xe không vé, giúp giảm đáng kể tỷ lệ tội phạm nghiêm trọng.
- Quản lý doanh nghiệp: Các công ty duy trì văn hóa kỷ luật nghiêm ngặt sẽ có năng suất làm việc cao hơn so với môi trường làm việc lỏng lẻo.

4. Ảnh hưởng xấu đến phát triển phần mềm

- Tích lũy nợ kỹ thuật: Các vấn đề nhỏ trong mã nguồn không được xử lý triệt để và thường xuyên sẽ dần trở thành vấn đề lớn.
- Giảm tính dễ đọc của mã: Mã không đồng nhất, chứa nhiều đoạn bị comment hoặc logic không rõ ràng sẽ làm khó hiểu hơn.

- Giảm khả năng của lập trình viên: Khi thấy mã nguồn lộn xộn, lập trình viên dễ có xu hướng tiếp tục làm theo những thói quen xấu đó làm cho ứng dụng ngày càng nhiều lỗi
- Tăng số lượng lỗi: Mã nguồn không được bảo trì tốt sẽ dễ mắc lỗi và lỗi chồng chất lên nhau.

5. Hướng giải quyết trong thiết kế phần mềm

- Refactor mã thường xuyên giúp cải thiện mã nguồn liên tục để loại bỏ những điểm còn sai và chưa hiệu quả.
- Code review thường xuyên sau khi hoàn thành một chức năng nào đó. Sử dụng công cụ kiểm tra mã tự động giúp phát hiện lỗi sớm để kịp thời loại bỏ và cải tiến.
- Khuyến khích thực hành mã sạch và đảm bảo rằng tất cả lập trình viên đều tuân theo các nguyên tắc viết mã tốt.
- Viết các test có thể kiểm tra các lỗi để đảm bảo hệ thống hoạt động ổn định.

II. The Boy Scout Rule

1. Giới thiệu

Quy tắc Boy Scout Rule (Hướng đạo sinh) trong lập trình phần mềm được Robert C. Martin (Uncle Bob) nhấn mạnh với câu nói nổi tiếng của Hội Hướng đạo sinh: "Hãy để nơi bạn đi qua sạch hơn lúc bạn đến".

2. Nội dung lý thuyết

- Quy tắc này có nguồn gốc từ hội Nam Hướng Đạo Hoa Kỳ, nơi các thành viên phải trả lại nơi cắm trại sạch hơn so với lúc họ tới.
- Điều này có nghĩa là họ cần xử lý những thứ họ đem đến tiện thể dọn dẹp một chút những thứ có sẵn ở đó để nơi nó sạch hơn so với khi họ đến.
- Trong lập trình, các lập trình viên không được làm cho mã nguồn cả nhóm tệ hơn sau khi mình code mà phải hỗ trợ cải thiện hơn các phần trước đó mà nhóm đã làm.

3. Ảnh hưởng xấu đến phát triển phần mềm

- Các đoạn code xấu hoặc các bản sửa lỗi nhanh làm giảm chất lượng của mã nguồn cơ bản.
- Khi không tuân thủ theo The Boy Scout Rule sẽ khiến cho nợ kỹ thuật tăng theo thời gian, khiến mã khó hiểu và khó cập nhật hơn.
- Trong các dự án nhóm, nhiều người làm việc trên cùng một mã. Lỗi của một người trở thành vấn đề của tất cả mọi người. Điều này dẫn đến hiểu lầm, lỗi và lãng phí thời gian.
- Nếu không giải quyết lỗi trong thời gian làm sẽ có lúc toàn bộ nhóm sẽ phải dừng lại để sửa lỗi và cải thiện mã nguồn. Điều này sẽ làm chậm deadline của ứng dụng khiến các bên liên quan cảm thấy khó chịu.

4. Hướng giải quyết trong thiết kế phần mềm

- Cải tiến mã nguồn: Khi chỉnh sửa hoặc thêm tính năng, hãy tối ưu hoặc sửa lỗi nhỏ trong phần liên quan. Khi chúng ta lập trình phần được giao hãy tranh thủ cải tiến các phần xung quanh đó.
- Refactor liên tục: Không để mã xấu tồn tại lâu, tránh để "cửa sổ vỡ" trong phần mềm.
- Code Review: Khuyến khích nhóm lập trình viên đánh giá và cải tiến mã lẫn nhau. Các lập trình viên cần mạnh dạn góp ý cho team để cải thiện thêm ứng dụng.
- Duy trì Clean Code: Viết mã dễ hiểu, có chú thích hợp lý và tránh lập code không cần thiết.

5. Lợi ích khi thực hiện The Boy Scout Rule

- Giảm nợ kỹ thuật: Bằng cách khắc phục sự cố nhanh nhất có thể, bạn sẽ ngăn ngừa được các vấn đề lớn hơn trong tương lai.
- Cải thiện mã nguồn: Khi lập trình viên thực hiện quy tắc này, mã nguồn sẽ ngày càng được tối ưu hóa, dễ đọc và dễ bảo trì hơn.
- Giảm được chi phí và thời gian bảo trì: Việc giảm nợ kỹ thuật, giúp tránh việc phải đại tu mã nguồn trong tương lai. Giảm thời gian debug và sửa lỗi do code được duy trì tốt.
- Hạn chế hiệu ứng “Cửa sổ vỡ” Ngăn chặn sự suy thoái của mã nguồn theo thời gian bằng cách cải thiện từng chút một.

III. Kết luận

- Cả hai lý thuyết này đều có giá trị trong thiết kế phần mềm và quản lý hệ thống. The Broken Window Theory giúp ngăn ngừa sự xuống cấp bằng cách xử lý lỗi sớm. Các lập trình viên không nên để xảy ra quá nhiều lỗi hổng trong chức năng mình đang làm, nếu thấy lỗi thì cần phải giải quyết nhanh nhất có thể.
- Trong khi The Boy Scout Rule khuyến khích sự cải tiến liên tục. Khi một lập trình viên thực hiện nhiệm vụ của mình thì phải cải tiến thêm các vấn đề xung quanh đó.
- Khi kết hợp cả hai nguyên tắc này, các nhóm phát triển phần mềm có thể tạo ra sản phẩm chất lượng cao, dễ bảo trì, ổn định và phát triển bền vững.