

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN**



**BỘ MÔN:
THIẾT KẾ PHẦN MỀM**

Bài tập 03 - The Broken Window Theory & The Boy Scout Rule

Giảng viên phụ trách	: Trần Duy Thảo
Đề tài	: Bài tập 03
Nhóm	: niceScore

Thành phố Hồ Chí Minh, tháng 3 năm 2025

MỤC LỤC

I. DANH SÁCH THÀNH VIÊN	2
II. NỘI DUNG.....	2
1. Giới thiệu chung	2
2. Thuyết cửa sổ vỡ.....	2
3. Quy tắc của Hướng đạo sinh	4
4. Tổng kết.....	4
III. TÀI LIỆU THAM KHẢO.....	5

I. DANH SÁCH THÀNH VIÊN

- MSSV: 22120062 – Nguyễn Đăng Điền
- MSSV: 22120186 – Huỳnh Tấn Lộc
- MSSV: 22120199 – Trần Lượng

II. NỘI DUNG

1. Giới thiệu chung

Trong đời sống nói chung và việc thiết kế phần mềm nói riêng, có nhiều nguyên tắc quan trọng mà chúng ta nên tuân theo để đem lại hiệu quả lớn nhất cho bản thân và xã hội, mọi người xung quanh. Trong số các nguyên tắc đó, có hai nguyên tắc đem lại lợi ích, góp phần xây dựng môi trường tốt hơn cho xã hội cũng như cộng đồng. Đó là Thuyết cửa sổ vỡ (“The broken window theory”), và Quy định của hướng đạo sinh (“The boy scout rule”). Cả hai đều nhấn mạnh vai trò của sự chú ý đến hành động nhỏ và trách nhiệm cá nhân trong việc duy trì một môi trường tốt hơn – một thành phố, một tổ chức hay một dự án phần mềm. Báo cáo sẽ tìm hiểu ý nghĩa của hai nguyên tắc này, cách chúng được áp dụng trong phát triển phần mềm, cũng như tác động của chúng đến chất lượng sản phẩm và tinh thần làm việc nhóm.

2. Thuyết cửa sổ vỡ

Thuyết cửa sổ vỡ xuất phát từ ngành tội học, được nêu lên lần đầu bởi James Q. Wilson và George L. Keilling năm 1982. Trọng tâm của học thuyết mô tả: Nếu một cửa sổ trong tòa nhà bị vỡ mà không được sửa chữa, những cửa sổ còn lại sẽ nhanh chóng bị phá vỡ theo. Giải thích cho hiện tượng, có nhiều nhân tố được đưa ra: sự tuân theo chuẩn mực xã hội; thiếu sự giám sát, quản lý; tội phạm mang tính cảnh báo (signal crimes). Một môi trường trật tự, sạch sẽ cho thấy khu vực này được giám sát thường xuyên; các hành vi phá hoại, không chuẩn mực sẽ dễ dàng bị phát hiện cũng như xử lý. Ngược lại, môi trường hỗn loạn phản ánh sự thiếu quan tâm, thờ ơ của cộng đồng; những hành động sai trái thường bị bỏ qua, ít khi bị phát hiện do thiếu quản lý. Theo thời gian, môi trường này càng xuống cấp, khiến việc sửa chữa, cải thiện trở nên khó khăn và kém ưu tiên. Điều này càng thu hút thêm hành vi tiêu cực, giảm động lực sửa chữa, duy trì của các thành viên trong cộng đồng.

Trong phát triển phần mềm, nếu xem môi trường là dự án, nhóm phát triển là cộng đồng thì

thuyết cửa sổ vỡ phản ánh rõ mối quan hệ giữa hành vi cá nhân với chất lượng dự án. Khi thêm một đoạn mã, một ghi chú, nếu ta không làm đúng theo quy chuẩn mà thêm một cách cầu thả, tác trách, điều này tương tự với làm vỡ một cửa sổ. Nếu hành vi không được phát hiện và sửa chữa nhanh chóng, các thành viên khác trong nhóm sẽ nhận ra sự thiếu quản lý của dự án và bắt chước để giảm bớt thời gian, công sức. Hệ quả của chuỗi hành động rất nghiêm trọng: dự án tràn ngập những đoạn mã yếu, tạm bợ, không đạt chuẩn. Điều này không chỉ gây khó khăn cho việc bảo trì, đặc biệt khi yêu cầu phần mềm thay đổi, mà còn ảnh hưởng đến tinh thần làm việc của nhóm. Dần dần, sự thiếu chuyên nghiệp lan rộng, dẫn đến mã nguồn kém chất lượng (*code smell*), khiến dự án trở nên khó sửa chữa, khó tái sử dụng, thậm chí thất bại hoàn toàn.

Dựa trên giả thuyết này, các chuyên gia đã đề xuất nhiều giải pháp nhằm giảm thiểu tác động tiêu cực. Đầu tiên, cần nâng cao chuẩn mực xã hội bằng cách thường xuyên nhắc nhở các thành viên về trách nhiệm duy trì môi trường trật tự, sạch sẽ. Khi mỗi cá nhân ý thức được vai trò của mình, họ sẽ có xu hướng hành động tích cực hơn. Thứ hai, tăng cường giám sát và phản hồi kịp thời. Việc phát hiện và khắc phục sớm những “cửa sổ vỡ” giúp ngăn chặn hiệu ứng lan rộng. Hoạt động này nên được thực hiện liên tục với sự tham gia của tất cả thành viên, thay vì chỉ phụ thuộc vào một nhóm nhỏ chịu trách nhiệm quản lý. Cuối cùng, cần có cơ chế nhắc nhở và xử lý các hành vi vi phạm. Điều này không chỉ giúp nâng cao nhận thức của cá nhân vi phạm mà còn tạo ra hiệu ứng răn đe trong cộng đồng, hạn chế tình trạng tái phạm và giữ vững tính kỷ luật chung.

Dựa theo các giải pháp trên, trong quá trình phát triển phần mềm, việc duy trì một mã nguồn sạch không chỉ phụ thuộc vào đội ngũ quản lý mà còn là trách nhiệm của từng thành viên trong dự án. Ngay từ khi khởi tạo, dự án cần được xây dựng với một nền tảng mã nguồn rõ ràng, có tổ chức, đồng thời các thành viên nên được nhắc nhở thường xuyên về tầm quan trọng của việc tuân thủ các tiêu chuẩn chất lượng. Khi mã nguồn được duy trì ngăn nắp và sạch sẽ, các lập trình viên có xu hướng cẩn trọng hơn, tránh việc đưa vào những đoạn code cầu thả, kém chất lượng. Bên cạnh đó, trách nhiệm không chỉ dừng lại ở phần mã của cá nhân, mà mỗi thành viên cũng cần giám sát và phản hồi về chất lượng mã nguồn của người khác. Việc này giúp nhanh chóng phát hiện mã nguồn xấu, tránh để chúng tích tụ và gây ảnh hưởng đến toàn bộ dự án. Các đoạn mã kém chất lượng cần được sửa chữa ngay lập tức, tránh tình trạng bỏ qua hoặc chấp nhận như một phần của hệ thống. Ngoài ra, thành viên chịu trách nhiệm cho phần mã không đạt chuẩn nên được xử lý phù hợp, nhằm đảm bảo tính kỷ luật và duy trì chất lượng chung của dự án, giữ thái độ chuyên nghiệp

cho nhóm phát triển.

3. Quy tắc của Hướng đạo sinh

Quy tắc của Hướng đạo sinh ban đầu được áp dụng cho môi trường tự nhiên, với nguyên tắc: "Luôn để lại nơi cắm trại sạch hơn lúc bạn tìm thấy nó." Ý tưởng cốt lõi của quy tắc là mỗi cá nhân đều có trách nhiệm cải thiện môi trường xung quanh, bất kể ai là người gây ra vấn đề, nhằm tạo điều kiện tốt hơn cho những người đến sau.

Trong lĩnh vực phát triển phần mềm, Quy định định của hướng đạo sinh có thể được hiểu là: "Luôn trả lại mã nguồn sạch hơn lúc bạn lấy nó". Điều quan trọng không phải là ai là tác giả ban đầu, mà là mỗi lập trình viên cần luôn nỗ lực cải thiện mã nguồn, cho dù là thay đổi nhỏ. Mã nguồn không nhất thiết phải hoàn hảo; nó chỉ cần tốt hơn một chút qua từng giai đoạn. Thay đổi có thể nhỏ như là xử lý bộ nhớ cuối chương trình, tách một hàm lớn ra nhiều hàm nhỏ, đổi tên biến phù hợp hơn, hay thêm ghi chú, ... Dần dần, mã nguồn sẽ ngày càng hoàn thiện, tạo ra một môi trường ngăn nắp, có tổ chức cho mọi thành viên khác, cũng như tạo thói quen lập trình tốt cho lập trình viên.

4. Tổng kết

Mặc dù Thuyết cửa sổ vỡ và Quy tắc Hướng đạo sinh có nguồn gốc khác nhau, cả hai đều nhấn mạnh tác động tích lũy của những hành vi nhỏ. Trong khi Thuyết cửa sổ vỡ cảnh báo về sự xuống cấp của môi trường xung quanh do hành vi tiêu cực không được kiểm soát, Quy tắc Hướng đạo sinh khuyến khích mỗi cá nhân chủ động cải thiện môi trường xung quanh. Trong phát triển phần mềm, hai nguyên tắc này bổ sung cho nhau: mã nguồn xấu có thể lan rộng nếu không được sửa chữa kịp thời, và cách để duy trì chất lượng là luôn để lại mã nguồn sạch hơn lúc lấy nó.

Trước đây, nhóm phát triển của chúng em thường gặp vấn đề với sự thiếu nhất quán trong mã nguồn, lỗi viết mã không đồng đều và các đoạn mã khó bảo trì. Những lỗi nhỏ thường bị bỏ qua, dẫn đến dự án ngày càng kém chất lượng. Sau khi tìm hiểu hai nguyên tắc trên, nhóm đã tổng kết và áp dụng một số kinh nghiệm quan trọng. Đầu tiên là xây dựng tiêu chuẩn chung ngay từ đầu để đảm bảo sự đồng nhất. Tiếp theo, mỗi cá nhân phải tuân thủ theo nguyên tắc chung và nhắc nhở các thành viên khác. Thứ ba, thường xuyên kiểm tra mã nguồn và nhanh chóng xử lý mã nguồn kém chất lượng. Cuối cùng, luôn cố gắng cải thiện mã nguồn, dù cho thay đổi có nhỏ nhất.

Cả hai nguyên tắc đều hướng đến việc duy trì trật tự và cải thiện chất lượng bằng cách tác động lên những hành vi nhỏ của cá nhân. Khi áp dụng vào phát triển phần mềm, hai nguyên tắc này giúp kiểm soát chất lượng mã nguồn, tránh sự xuống cấp và thúc đẩy văn hóa làm việc chuyên nghiệp trong nhóm phát triển. Một dự án phần mềm thành công không chỉ đến từ thiết kế tốt, mà còn từ tư duy duy trì và cải thiện liên tục - đó chính là giá trị cốt lõi mà hai nguyên tắc mang lại.

III. TÀI LIỆU THAM KHẢO

- James Q. Wilson & George L. Kelling, *Broken Windows: The police and neighborhood safety*, The Atlantic, 1982.
<https://www.theatlantic.com/magazine/archive/1982/03/broken-windows/304465/>
- Broken Window Theory: https://en.wikipedia.org/wiki/Broken_windows_theory
- The Boy Scout Rule: https://97-things-every-x-should-know.gitbooks.io/97-things-every-programmer-should-know/content/en/thing_08/