

Nợ kỹ thuật (Technical debt) (Nguyên nhân xảy ra)

Vấn đề

- Mọi lập trình viên đều cố gắng viết code sạch từ đầu. Nhưng đến lúc nào thì code trở nên không sạch?

Ví dụ: Nếu bạn vay tiền từ ngân hàng, điều này cho phép bạn mua hàng nhanh hơn. Nhưng phải trả thêm tiền lãi để đẩy nhanh quá trình. Nếu tích lũy tiền lãi đến mức số tiền lãi vượt qua khối thu nhập của bạn, việc trả nợ trở nên bất khả thi.

=> Tương tự code, tạm thời có thể tăng tốc mà không cần phải viết test cho các feature. Nhưng điều này sẽ dần dần làm chậm tiến độ mỗi ngày cho đến khi bạn trả hết nợ bằng cách viết tests.

Nguyên nhân

- Áp lực kinh doanh:
 - Hoàn cảnh kinh doanh buộc phải triển khai tính năng trước khi hoàn thành
 - Trường hợp này, các bản vá và vá lỗi sẽ xuất hiện trong code để ẩn các phần chưa hoàn thiện của dự án
- Thiếu hiểu biết về hậu quả của nợ kỹ thuật
- Không chống lại được sự liên kết chặt chẽ của các thành phần
 - Dự án đôi khi giống 1 khối thống nhất hơn là các module riêng lẻ.
 - Bất kỳ thay đổi nào đối với một phần của dự án sẽ ảnh hưởng đến các phần khác
- Thiếu các bài tests
 - Những thay đổi được triển khai và đưa vào sản xuất mà không được test. Hậu quả rất thảm khốc
 - Ví dụ: 1 bản sửa lỗi nóng có vẻ vô hại có thể gửi 1 email thử nghiệm đến hàng nghìn user hoặc thậm chí làm hỏng db
- Thiếu tài liệu
 - Làm chậm quá trình giới thiệu của người mới vào dự án.
 - Chậm quá trình phát triển nếu người chủ chốt rời khỏi dự án.
- Thiếu đi sự tương tác giữa các thành viên trong nhóm
 - Nếu cơ sở kiến thức không được phân phối trong toàn công ty, mọi người sẽ làm việc với sự hiểu biết lỗi thời về quy trình và thông tin về dự án.
 - Tình hình có thể trầm trọng hơn khi dev mới vào nghề được cố vấn

đào tạo không đúng cách.

- Phát triển đồng thời trong nhiều nhánh dài hạn
 - Là dev làm việc trên nhiều nhánh khác nhau của code cùng một lúc trong thời gian dài
 - Dẫn đến sự tích tụ nợ kỹ thuật, sau đó tăng lên khi các nhánh được merge
 - Càng có nhiều thay đổi thì nợ kỹ thuật càng lớn
- Trì hoãn refactoring (cấu trúc)
 - Các yêu cầu của dự án liên tục thay đổi và đến 1 lúc nào đó, có thể thấy rõ ràng 1 số phần của code đã lỗi thời, cồng kềnh và cần phải code lại để đáp ứng các yêu cầu mới
 - Mặt khác, dev đang viết mã mới mỗi ngày để làm việc vs các phần lỗi thời.
 - Dẫn đến việc tái cấu trúc càng bị trì hoãn thì sẽ càng phải làm lại mã phụ thuộc nhiều hơn trong tương lai.
- Thiếu giám sát tuân thủ
 - Xảy ra khi mọi người làm việc trong dự án viết mã theo cách họ thấy phù hợp
 - Không theo convention code
- Trình độ kỹ thuật
 - Dev thiếu trình độ và không biết viết code tốt