Chương 2: Nguyên tắc refactor code

Refactor code là quá trình tái cấu trúc lại mã nguồn nhằm cải thiện chất lượng hệ thống, tăng khả năng bảo trì và tối ưu hiệu năng mà **không làm thay đổi hành vi nghiệp vụ của hệ thống**. Để đảm bảo việc refactor diễn ra an toàn, hiệu quả và không làm phát sinh lỗi mới, cần tuân thủ một số nguyên tắc cơ bản trong quá trình thực hiện.

- 2.1. Không thay đổi hành vi nghiệp vụ của hệ thống
- 2.2. Thực hiện refactor theo từng bước nhỏ
- 2.3. Đảm bảo khả năng kiểm thử sau khi refactor

2.1. Không thay đổi hành vi nghiệp vụ của hệ thống

Nguyên tắc quan trọng nhất của refactor là **không làm thay đổi logic nghiệp vụ hoặc kết quả xử lý của hệ thống**. Mọi thay đổi trong quá trình refactor chỉ nhằm cải thiện cấu trúc code, giảm độ phức tạp và tăng hiệu năng.

Trước và sau khi refactor, API phải đảm bảo:

- Trả về cùng một cấu trúc dữ liệu

- Giữ nguyên logic xử lý nghiệp vụ

- Không làm thay đổi dữ liệu hệ thống

2.2. Thực hiện refactor theo từng bước nhỏ

Refactor nên được thực hiện theo **các thay đổi nhỏ, có kiểm soát**, thay vì thay đổi lớn trong một lần. Điều này giúp dễ dàng kiểm tra và phát hiện lỗi.

Quy trình thường bao gồm:

1. Xác định vấn đề trong code
2. Refactor một phần nhỏ của hệ thống
3. Kiểm thử lại chức năng
4. Đánh giá hiệu năng
5. Tiếp tục refactor các phần tiếp theo

Cách tiếp cận này giúp giảm rủi ro và đảm bảo hệ thống luôn hoạt động ổn định trong quá trình cải tiến.

2.3. Đảm bảo khả năng kiểm thử sau khi refactor

Sau mỗi lần refactor, cần thực hiện kiểm thử để đảm bảo hệ thống vẫn hoạt động đúng như trước. Việc kiểm thử có thể bao gồm:

- Kiểm thử chức năng API
- So sánh dữ liệu trả về trước và sau refactor
- Kiểm tra thời gian xử lý API
- Kiểm tra log hệ thống

Kiểm thử giúp phát hiện sớm các lỗi có thể phát sinh trong quá trình tái cấu trúc code.