

UNIT TEST COVERAGE & BEST PRACTICE

NHÓM 22: TTT
PHAN VĂN TÀI - 22120316
ĐINH ĐỨC THIÊN - 22120343
VÕ KHẮC THIÊN - 22120345

CÁC LOẠI CODE COVERAGE QUAN TRỌNG



Line Coverage

Là chỉ số cơ bản và phổ biến nhất để đo lường mức độ kiểm thử của một phần mềm. Tỷ lệ này tính toán số lượng dòng mã được thực thi so với tổng số dòng mã trong ứng dụng.



Branch Coverage

Đo lường việc kiểm thử tất cả các nhánh trong chương trình, đảm bảo rằng tất cả các lựa chọn trong logic đều được kiểm tra.



Function Coverage

Tập trung vào việc kiểm tra tất cả các hàm, method hoặc chức năng được gọi trong mã nguồn.



Path Coverage

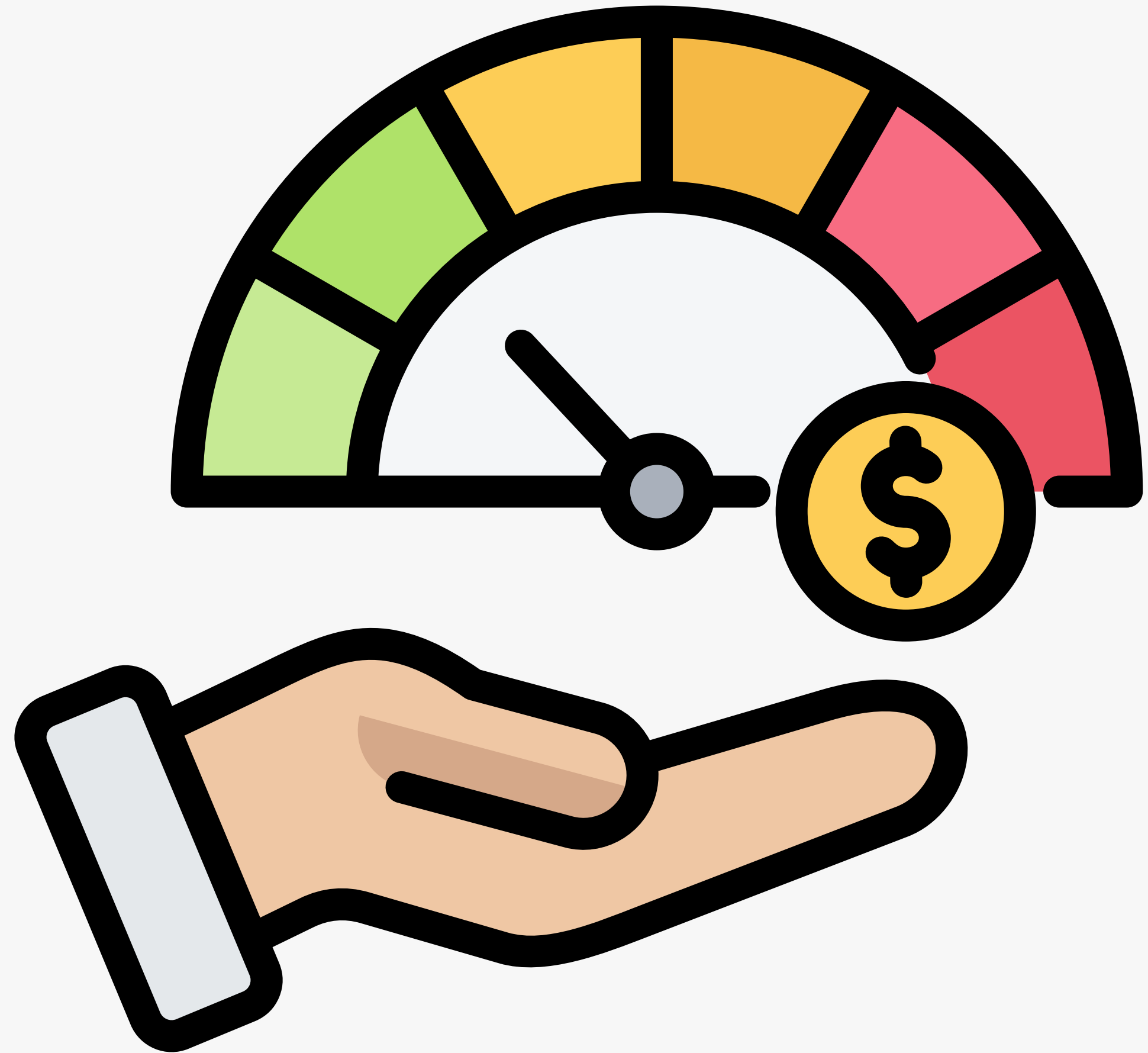
Đảm bảo rằng tất cả các đường đi logic quan trọng đều được kiểm thử.

MỨC COVERAGE TỐI THIỂU

Một chuẩn chung thường là từ 70% đến 90% coverage cho các dòng mã.

Nếu phần mềm là hệ thống nhúng hoặc cần độ tin cậy cao (ngân hàng, chăm sóc sức khỏe), yêu cầu lên đến 100%.

Các đoạn mã xử lý logic chính và các phần có thể gây lỗi nghiêm trọng cần được kiểm thử đầy đủ.



LỢI ÍCH CỦA COVERAGE ĐÚNG MỨC



Giảm thiểu rủi ro

Test coverage giúp phát hiện lỗi trong quá trình phát triển, tránh các lỗi nghiêm trọng khi triển khai.



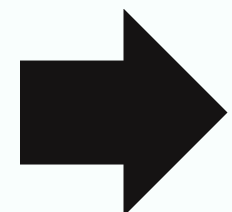
Giúp dễ dàng bảo trì

Một bộ unit test tốt giúp bảo vệ các phần quan trọng của hệ thống khi có thay đổi.



Giảm thời gian kiểm thử thủ công

Khi có coverage tốt, bạn không cần phải kiểm thử thủ công quá nhiều mà có thể tin tưởng vào các unit test tự động.



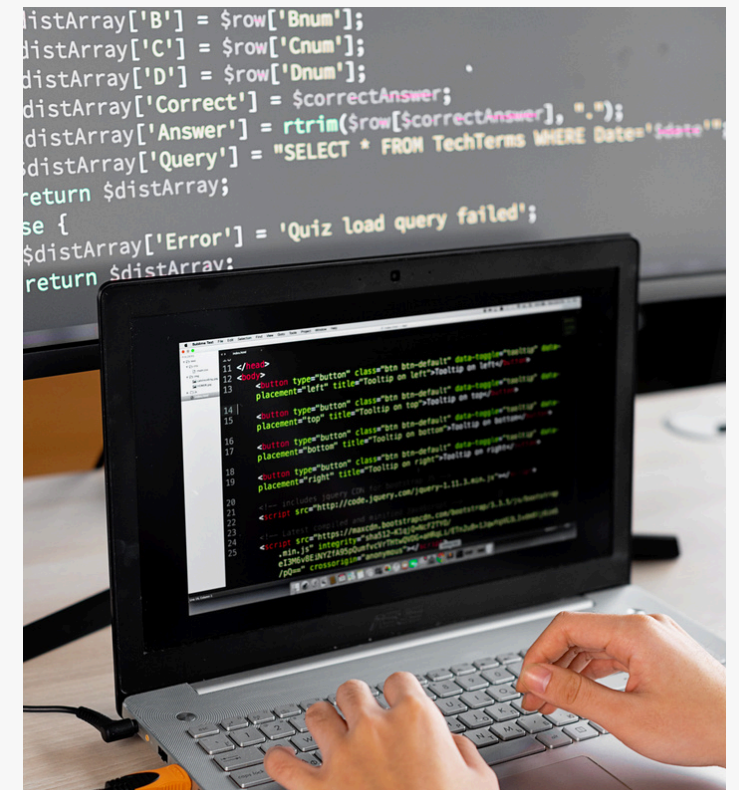
Đừng chỉ chạy theo con số coverage cao nhất. Quan trọng nhất là phải đảm bảo rằng các phần mã quan trọng và các kịch bản sử dụng chính đều được kiểm thử kỹ lưỡng.

BEST PRACTICE KHI VIẾT UNIT TEST

Viết Test Dễ Bảo Trì: Hãy tạo ra các unit test có thể dễ dàng bảo trì và mở rộng. sử dụng mocking để giả lập các yếu tố bên ngoài đó.

Happy Case: Là tình huống mà tất cả các đầu vào đều hợp lệ.

Edge Case: Là các tình huống biên giới hoặc đặc biệt mà người dùng có thể ít khi gặp



BEST PRACTICES (TT)

Không Viết Test Trùng Lặp:

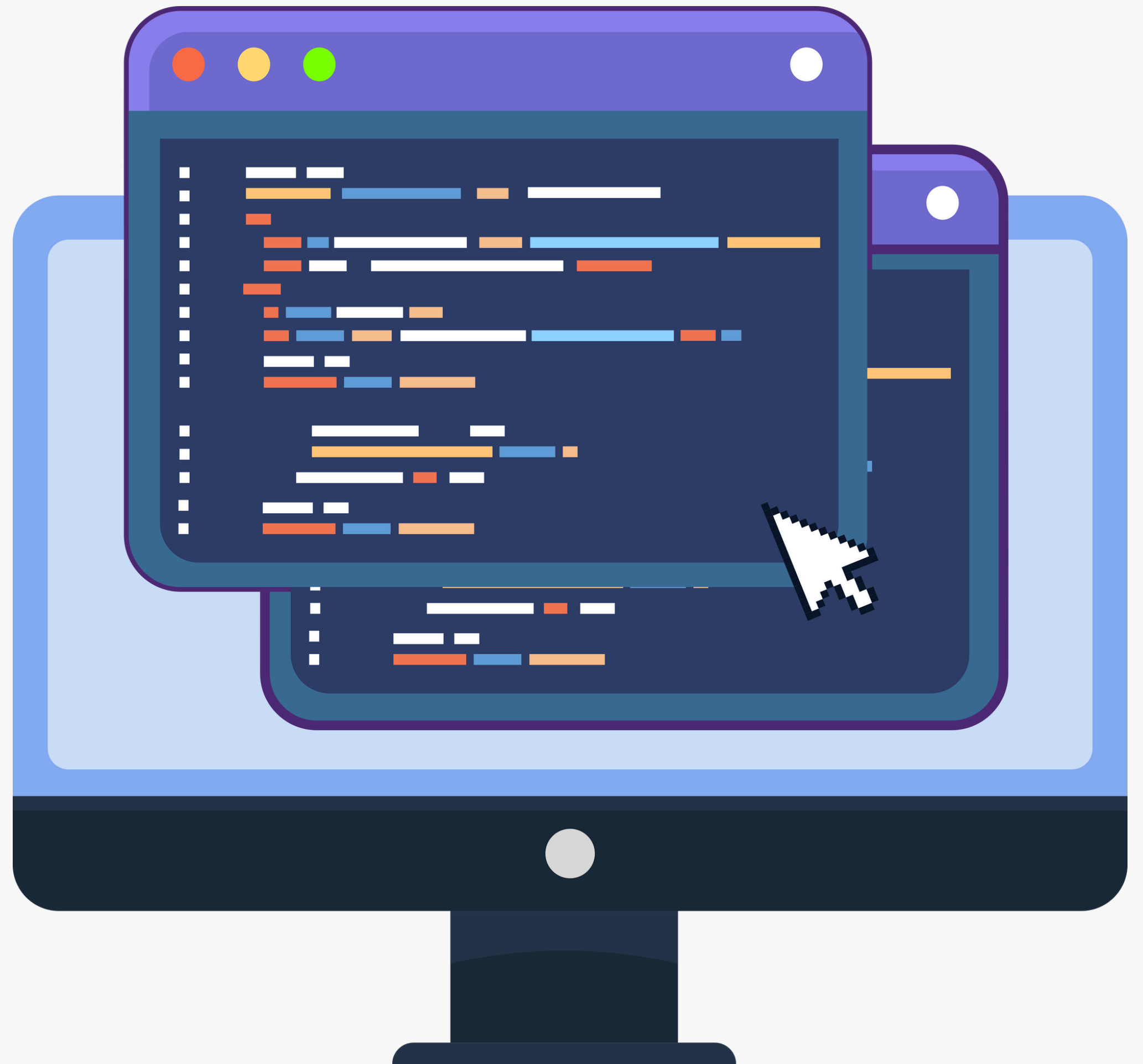
Hãy chắc chắn rằng mỗi test chỉ kiểm tra một tình huống cụ thể và không quá phụ thuộc vào cách mà mã được triển khai.

Tối ưu hóa Test Coverage:

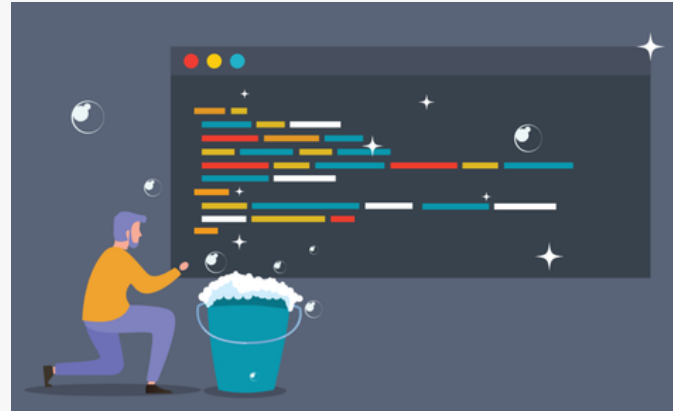
Thực hiện kiểm thử tự động cho các trường hợp cơ bản và các trường hợp biên.

Viết Test Không Phụ Thuộc Vào Implementation:

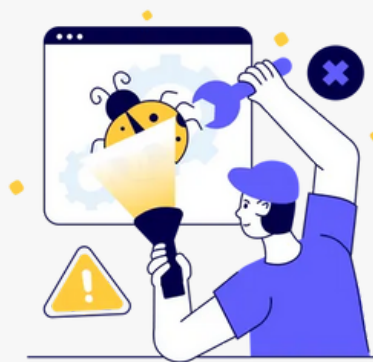
Viết các test có tính chất blackbox



KẾT LUẬN



Đảm bảo đầy đủ các yếu tố coverage



Đảm bảo coverage đủ để kiểm thử các logic quan trọng



Viết unit test dễ bảo trì, kiểm thử cả happy case và edge case, tránh trùng lặp



Hiểu rõ mục tiêu và phương pháp kiểm thử