

**Họ tên:** Phạm Gia Việt Anh

**Lớp:** INT3117 1

**Mã sinh viên:** 21020051

Đề bài: Tìm hiểu ưu điểm và nhược điểm của kiểm thử tích hợp theo phương pháp top-down và bottom-up.

Bài làm:

Với phương pháp top-down, các ưu điểm được thể hiện là:

- **Kiểm tra các giao diện trước:** Vì kiểm thử sẽ được tính hợp từ hàm chính – gốc của cây, nên khả năng cao hàm chính này sẽ liên quan trực tiếp đến giao diện người dung (nếu phần mềm có). Từ đó giúp kiểm tra được giao diện người dung trước để đảm bảo tính hoạt động tốt và trải nghiệm người dùng tốt.
- **Tính sử dụng lại:** Các testcase khi sử dụng để kiểm tra sự tích hợp giữa các module có thể được sử dụng lại để kiểm thử hồi quy.
- **Phát hiện sai sót:** Nếu xuất hiện sai sót từ phần đầu chương trình, ta có thể dễ dàng kiểm tra được. Hay nói cách khác, các module quan trọng từ đầu chương trình sẽ được kiểm tra đầu tiên.
- **Lỗi thiết kế:** Vì các module ở dưới có thể thay thế bằng stubs, do đó phương pháp này cung cấp module hoạt động sớm của chương trình nên có thể phát hiện được lỗi thiết kế sớm và sửa sớm.
- **Bản mẫu:** Với phương pháp này, có thể sử dụng bản mẫu ngay từ đầu và dần dần thay thế thành các module hoàn chỉnh.

Các nhược điểm của phương pháp top-down là:

- **Viết hàm giả:** Việc viết các stubs ở các module nhỏ hơn là rất nhiều. Do đó sẽ tốn thời gian và công sức. Ngoài ra, cần quản lý chúng tốt để có thể chuyển giữa mã giả và mã thật, do mã giả và thật không thể cùng tồn tại.

- **Sự phụ thuộc:** Phương pháp top-down yêu cầu các thành phần chức năng ở mức cao phải được triển khai hoàn toàn hoặc được thay thế bằng các đối tượng giả lập để kiểm tra. Điều này có thể dẫn đến sự phụ thuộc và chờ đợi.
- **Lựa chọn testcase:** Việc lựa chọn testcase để thiết kế stubs càng trở nên khó khăn khi các stubs càng cách xa các top-level module.

Với phương pháp bottom-up, các ưu điểm được thể hiện là:

- **Tập trung vào chức năng cụ thể:** Phương pháp bottom-up cho phép bạn kiểm tra và hoàn thiện từng chức năng cụ thể ở các module nhỏ đến lớn, trước khi tích hợp chúng, giúp tập trung vào từng phần riêng lẻ.
- **Viết các testdriver:** Vì cây thường có xu hướng xoè ra ở lá, nên các hàm testdriver sẽ ít hơn các hàm stubs trong thực tế.
- **Quan sát:** Việc tích hợp các module từ dưới lên giúp cho việc quan sát kết quả của các chức năng dễ dàng hơn, từ đó việc phát hiện ra vị trí các lỗi sai sẽ dễ hơn.
- **Tiến độ:** Việc kiểm tra các module nhỏ trước đảm bảo quá trình kiểm thử vẫn diễn ra dù các module cấp cao vẫn đang được phát triển.

Nhược điểm của phương pháp bottom-up:

- **Không kiểm tra tích hợp toàn diện sớm:** Bottom-up không kiểm tra toàn bộ tích hợp của hệ thống cho đến khi tất cả các thành phần đã hoàn thiện, điều này có thể làm trễ phát hiện lỗi tích hợp.
- **Bản mẫu:** Xây dựng 1 bản mẫu ngay từ đầu là điều không thể.
- **Phức tạp:** Nếu hệ thống có nhiều module con thì việc làm như thế này sẽ tốn thời gian.
- **Tính quan sát:** Không thể quan sát các chức năng ở mức độ hệ thống cho đến khi có testdriver.
- **Testdriver:** Việc viết testdriver sẽ phức tạp hơn so với việc viết 1 stub.

- **Tiến độ:** Không thể có sản phẩm mẫu trước khi testdriver cuối cùng được hoàn thành.

Chọn phương pháp kiểm thử tích hợp phụ thuộc vào dự án cụ thể, sự phát triển của phần mềm, và yêu cầu cụ thể của dự án. Trong nhiều trường hợp, pha kiểm thử tích hợp có thể sử dụng cả hai phương pháp để đảm bảo tính nhất quán và đáng tin cậy của hệ thống.