

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



UNIT TEST COVERAGE

THIẾT KẾ PHẦN MỀM

22CLC09

Nguyễn Quang Minh 22127270

I. Tổng quan

- Unit test coverage là một chỉ số đo lường mức độ các đoạn mã của ứng dụng được kiểm tra bởi các unit test. Trong môi trường công nghiệp, việc nghiên cứu và áp dụng các tiêu chuẩn unit test coverage không chỉ giúp đảm bảo chất lượng phần mềm mà còn tăng cường khả năng duy trì và phát triển hệ thống.

II. Các loại Code Coverage Quan trọng:

1 Statement Coverage:

- Đo lường tỉ lệ số dòng lệnh được thực thi khi chạy các bài kiểm tra. Đây là chỉ số cơ bản nhất, nhưng có thể bỏ sót các trường hợp rẽ nhánh của logic.

2 Branch Coverage:

- Đánh giá việc thực thi tất cả các nhánh điều kiện (if/else, switch-case). Điều này giúp kiểm tra các tình huống logic phức tạp.

3 Condition Coverage:

- Tập trung đánh giá các điều kiện Boolean trong các biểu thức. Mục tiêu là đảm bảo mỗi điều kiện có giá trị đúng và sai được kiểm tra.

4 Path Coverage:

- Kiểm tra tất cả các đường đi có thể của chương trình. Đây là mục tiêu lý tưởng nhưng khó thực hiện đối với các ứng dụng lớn do số lượng đường đi có thể là vô tận.

5 Function/Method Coverage:

- Đo lường tỉ lệ hàm, phương thức được gọi trong quá trình kiểm thử

III. Tiêu chuẩn Coverage trong Industry:

- Ngưỡng Coverage:
 - Rất nhiều tổ chức đặt mục tiêu khoảng 80% statement coverage làm tiêu chuẩn tối thiểu. MT s d án quan trng có th t mc tiêu cao hn (90%-100%) gim thiu ri ro.
 - Tuy nhiên chỉ số cao không đồng nghĩa với chất lượng kiểm thử tốt. Một bộ test có thể đạt mức 90% coverage nhưng vẫn bỏ sót những lỗi nghiêm trọng nếu các trường hợp biên chưa được kiểm tra đầy đủ.
- Chất Lượng Bài Test:
 - Tính độc lập: Các Unit test nên được thiết kế để kiểm tra từng thành phần riêng biệt mà không phụ thuộc vào trạng thái chung của hệ thống.
 - Tính dễ bảo trì: Khi mã nguồn thay đổi, Unit Test cũng cần được cập nhật để phản ánh những thay đổi đó.
 - Sử dụng mocks và Stubs: Giúp cô lập các Module, tập trung kiểm tra logic nghiệp vụ.
- Kiểm Tra Các Tình Huống Biên:
 - Ngoài việc đo lường coverage, các bài test nên đảm bảo kiểm tra các trường hợp biên, các ngoại lệ và các trường hợp lỗi có thể xảy ra.

IV. Công Cụ và Công Nghệ Hỗ Trợ:

- Trong thực tế, có nhiều công cụ giúp đo lường Unit Test Coverage như:
 - JaCoCo (Java): Đo lường Coverage cho các dự án Java.
 - Vitest (JavaScript/TypeScript): Công cụ phổ biến cho các ứng dụng Web.
 - Coverage.py (Python): Hỗ trợ đo lường Coverage cho Python.
 - Visual Studio Code Coverage Tools (C/.NET): Được tích hợp sẵn trong môi trường phát triển.

V. Thách thức và Lưu Ý:

- Coverage không bằng chất lượng test: Một chỉ số cao không đảm bảo rằng mọi lỗi đều được phát hiện. Các bài test cần phải có tính kiểm tra logic đúng đắn và đảm bảo rằng các tình huống bất thường được xử lý.
- Phát sinh giả mạo: Có thể xảy ra trường hợp các bài test được viết chỉ để "điền số" coverage, dẫn đến kiểm thử không thực sự hiệu quả. Do đó, cần đánh giá chất lượng của từng test case.
- Bảo trì và cập nhật: Khi hệ thống thay đổi, Unit Test cũng cần được cập nhật để đảm bảo chúng vẫn phù hợp với phiên bản hiện tại của mã nguồn.

VI. Tổng kết:

- Trong môi trường công nghiệp, Unit Test Coverage là một chỉ số quan trọng để đảm bảo chất lượng phần mềm, nhưng không nên xem nó như tiêu chí duy nhất. Các tổ chức thường đặt ngưỡng từ 80% trở lên tùy thuộc vào độ phức tạp và mức độ rủi ro của thuật dự án. Quan trọng hơn cả là cách thiết kế các bài test sao cho phản ánh chính xác logic nghiệp vụ và xử lý các tình huống bất thường một cách hiệu quả.
- Việc nghiên cứu và áp dụng các tiêu chuẩn này không chỉ giúp cải thiện chất lượng sản phẩm mà còn hỗ trợ cho quá trình bảo trì, phát triển và mở rộng hệ thống trong tương lai.