

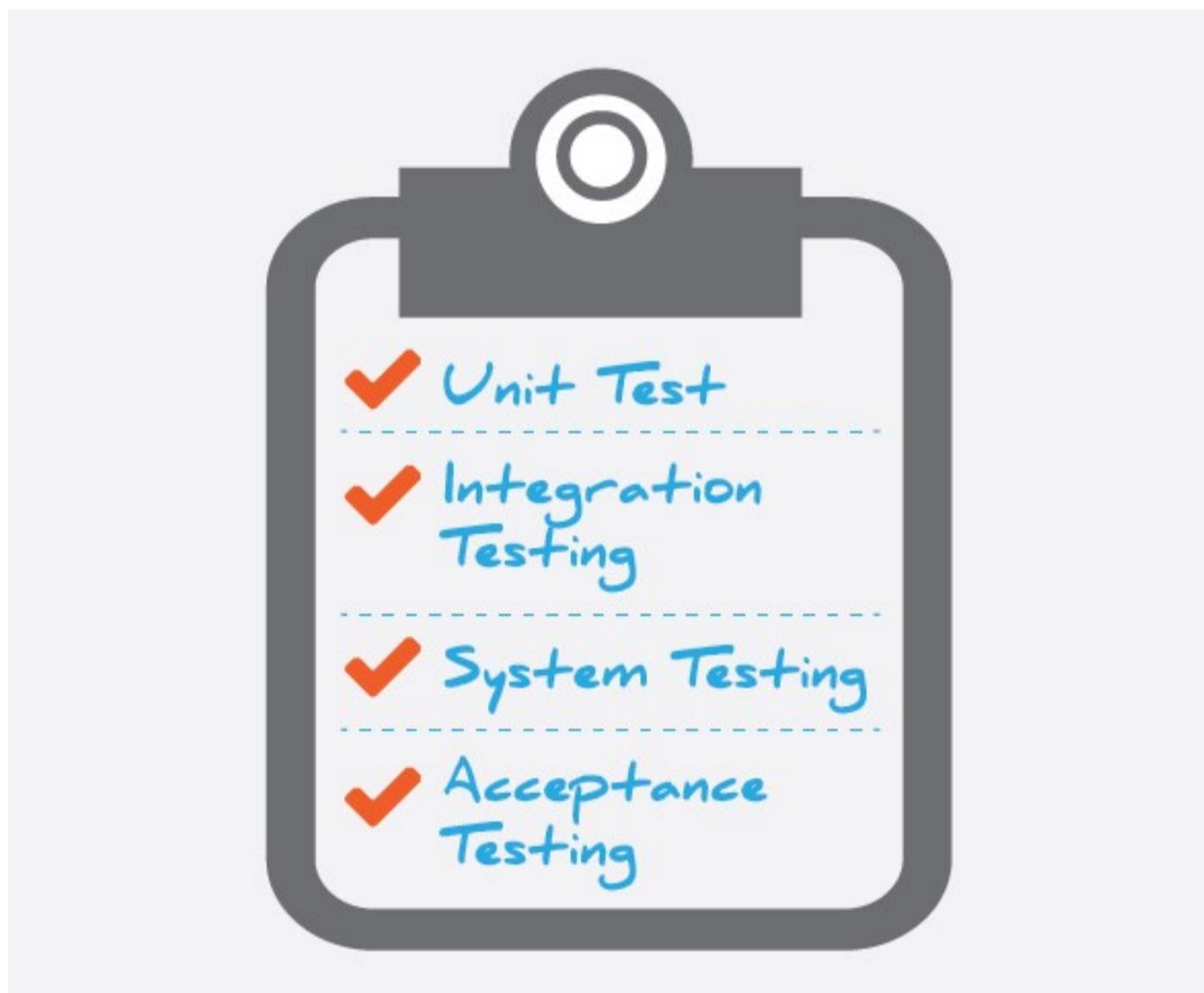
# Level Test

Tags

- [1. Unit Testing \(Kiểm thử đơn vị\)](#)
- [2. Integration Testing \(Kiểm thử tích hợp\)](#)
- [3. System Testing \(Kiểm thử hệ thống\)](#)
- [4. Acceptance Testing \(Kiểm thử chấp nhận\)](#)

Trước khi release một phần mềm nào đó, chúng đã phải trải qua một quá trình kiểm tra kỹ lưỡng để đảm bảo rằng phần mềm sẽ hoạt động mượt mà, ổn định theo đúng chức năng được thiết kế. Có **04 mức độ kiểm thử phần mềm** chính cần hoàn thành trước khi đưa phần mềm vào sử dụng: unit testing, integration testing, system testing, và acceptance testing.

Nhiều bạn sẽ đặt câu hỏi tại sao không bao gồm Regression Tesing (Kiểm thử hồi quy)? Kiểm thử hồi quy không phải là một cấp độ kiểm thử riêng biệt; nó chỉ là một loại kiểm thử có thể được thực hiện trong bất kỳ giai đoạn nào trong bốn mức độ kiểm thử phần mềm chính mà CO-WELL đã nhắc đến ở trên.



## 1. Unit Testing (Kiểm thử đơn vị)

Unit testing, còn được gọi là Component testing, mức độ kiểm thử phần mềm này thường do Developer phụ trách, họ sẽ đi kiểm tra các module, các hàm, các phương thức, các lớp,... mà họ viết ra nhằm gia tăng sự tin cậy cho các chức năng mà mình viết.

Kiểm thử đơn vị nằm trong phạm vi của Kiểm thử hộp trắng (White-box testing), tức là kiểm tra code bên trong của một chức năng hoặc hệ thống để xem chức năng hoặc hệ thống đó được viết đúng chuẩn code hay chưa, đoạn code đó khi chạy hiệu năng có tốt hay không, có nhanh hay không, có tốn tài nguyên hay không,..

Tuy nhiên, kiểm thử theo phương pháp này không thể tìm được mọi lỗi của ứng dụng. Không thể đánh giá mọi tình huống có thể xảy ra trong chương trình. Nó bị giới hạn bởi các tình huống và bộ dữ liệu test mà developer sử dụng để kiểm thử

chương trình.

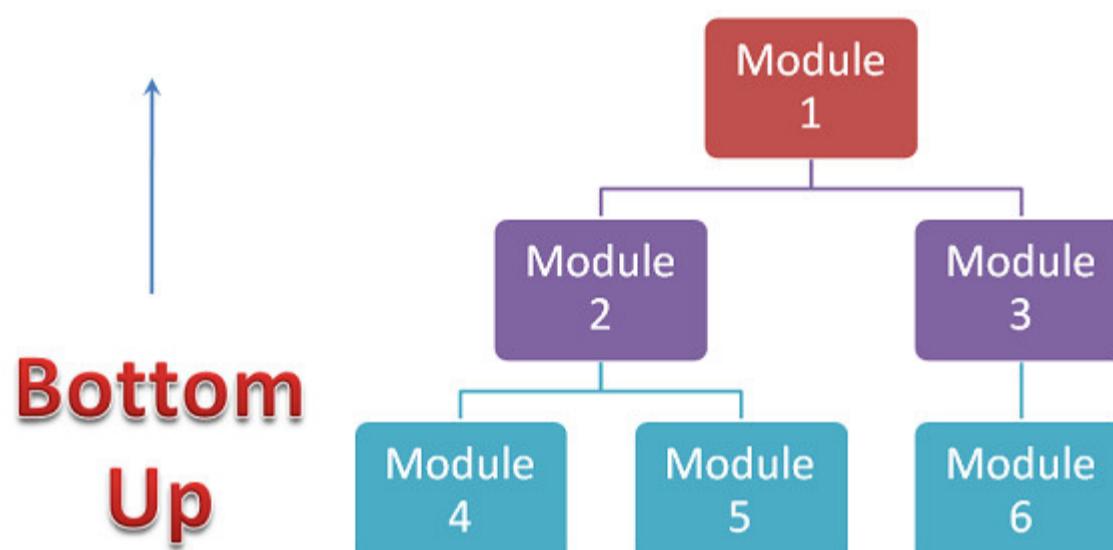
## 2. Integration Testing (Kiểm thử tích hợp)

Kiểm thử tích hợp là kiểm thử sự tương tác giữa các chức năng với nhau trong hệ thống và được thực hiện bởi Tester. Ví dụ: sau khi đã unit test chức năng đăng nhập và chức năng đăng ký thì ta có thể tiến hành kiểm thử tích hợp của 2 chức năng này để xem chúng có tương tác tốt với nhau không, sau khi đăng ký thành công thì ta có thể tiến hành đăng nhập bằng tài khoản đã đăng ký xem có thực hiện được không.

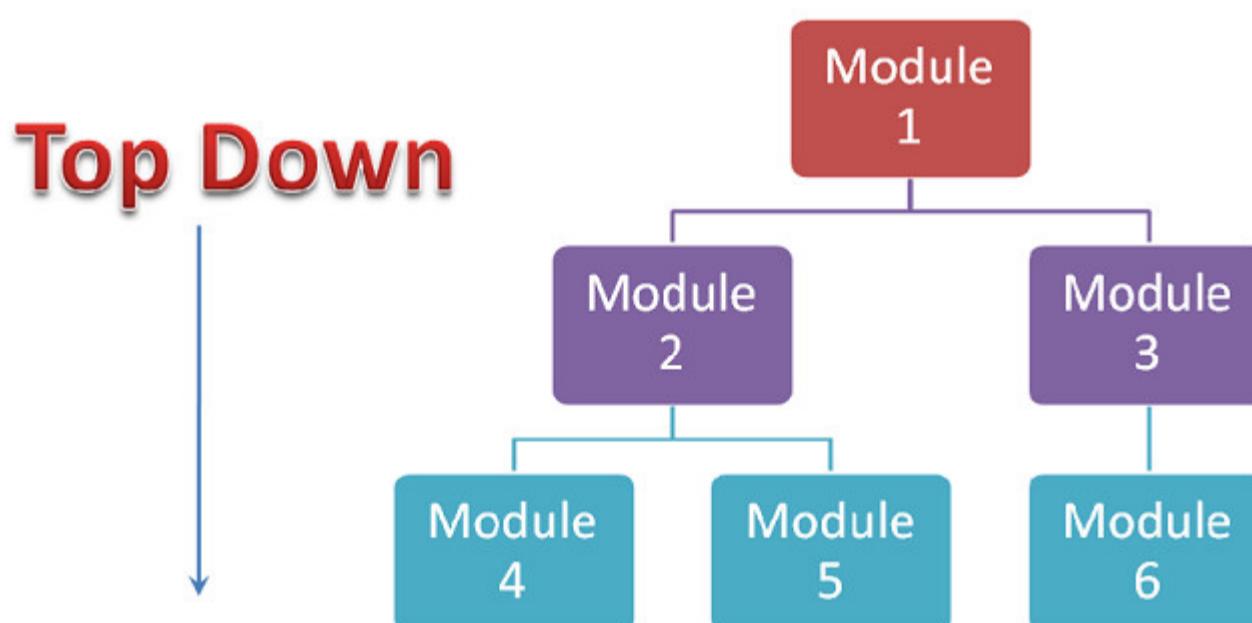
Một ví dụ khác: Sau khi unit test cho các chức năng con trong chức năng đăng ký môn học như đăng nhập, đăng ký tài khoản, kiểm tra điều kiện tiên quyết, kiểm tra học phí,... thì ta kiểm tra sự tích hợp giữa các chức năng này bằng cách tiến hành đăng ký một môn học để xem sự tương tác giữa các chức năng này có thực hiện được không, có trơn tru không, có bị mất liên kết chỗ nào không,...

Integration Testing có hai cách tiếp cận phổ biến: Tích hợp từ dưới đi lên (Bottom-up integration) và tích hợp đi trên đi xuống (Top-down integration)

**Tích hợp từ dưới đi lên:** Việc kiểm thử được bắt đầu bằng unit test, sau đó sẽ đi đến mức cao hơn, đó là kết hợp các thành phần và chức năng của hệ thống.



**Tích hợp từ trên đi xuống:** Những module cấp cao nhất được thử nghiệm đầu tiên rồi đến các module thấp hơn được thử nghiệm.

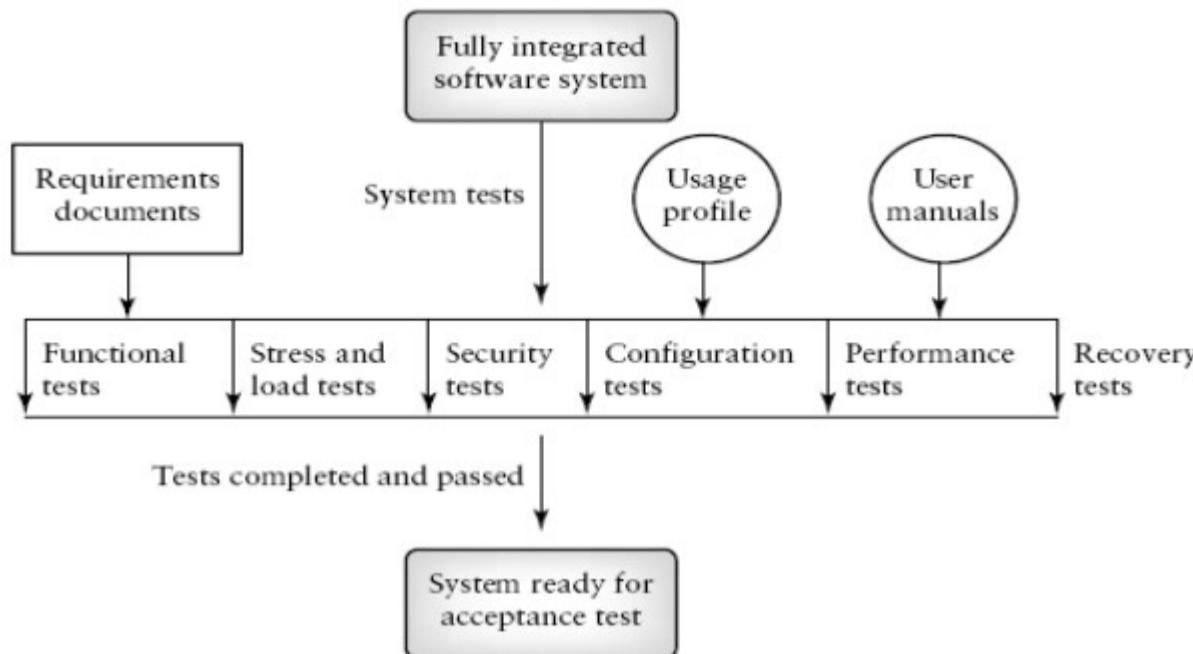


Trong một môi trường phát triển phần mềm hiện đại, kiểm thử từ dưới lên thường được thực hiện đầu tiên và sau đó là kiểm thử từ trên xuống. Quá trình này được kết thúc sau nhiều vòng kiểm thử trên ứng dụng hoàn chỉnh, tốt nhất là sử dụng các kịch bản kiểm thử dựa trên các tình huống thực tế.

## 3. System Testing (Kiểm thử hệ thống)

Kiểm thử hệ thống là kiểm thử một hệ thống đã hoàn thành, đã tích hợp đầy đủ các chức năng nhằm kiểm tra xem hệ thống phần mềm đó có đáp ứng đầy đủ các yêu cầu chức năng theo bản đặc tả yêu cầu phần mềm (SRS) hay không. Người thực hiện test level này thường là Tester.

Kiểm thử hệ thống thuộc phạm vi Kiểm thử hộp đen (tức là Tester chỉ quan tâm đầu vào và kết quả mong đợi ở đầu ra mà không cần kiểm tra code bên trong được viết như thế nào).



Dưới đây là một số loại kiểm thử thường được thực hiện trong System Test:

- Kiểm thử chức năng (**Functional Test**): Là kiểm thử toàn bộ hệ thống, đảm bảo hệ thống hoạt động đúng theo yêu cầu được đưa ra trước đó
- Kiểm thử hiệu năng (**Performance Test**): Là kiểm tra sự tuân thủ của hệ thống với các yêu cầu được chỉ định về hiệu năng. Xác định những thuộc tính chất lượng của hệ thống như khả năng mở rộng, độ tin cậy...
- Kiểm thử cơ sở dữ liệu (**Database Test**): Là kiểm tra dữ liệu hiển thị trên hệ thống có giống với dữ liệu trong cơ sở dữ liệu hay không?
- Kiểm thử khả năng bảo mật (**Security Test**): Là kiểm tra hệ thống được bảo vệ an toàn, không bị đánh cắp dữ liệu, thông tin trước các tấn công từ bên ngoài.
- Kiểm thử tính khả dụng (**Usability Test**): Kiểm tra tính thân thiện với người dùng và tính dễ sử dụng của hệ thống.
- Kiểm tra tính tương thích (**Compatibility Test**): Là kiểm tra xem hệ thống có tương thích với các yếu tố khác của hệ thống mà nó sẽ hoạt động hay không? (Ví dụ: Trình duyệt, hệ điều hành, phần cứng).
- Kiểm tra khả năng phục hồi (**Recovery Test**): Là kiểm tra hệ thống có khả năng khôi phục trạng thái ổn định khi gặp các sự cố bất thường không.
- 

## 4. Acceptance Testing (Kiểm thử chấp nhận)

Mức độ kiểm thử phần mềm cuối cùng chính là Acceptance Test (Kiểm thử chấp nhận) – kiểm tra xem hệ thống có đáp ứng đúng nhu cầu và mong đợi của khách hàng hay không.

Kiểm thử chấp nhận thường là trách nhiệm của người dùng hoặc khách hàng. Trong kiểm thử hệ thống, khách hàng sẽ kiểm tra xem phần mềm được viết có hoạt động đúng như mong đợi của mình không, có đảm bảo tính tiện dụng, hiệu suất hoạt động có như mong đợi không, có bảo mật tốt hay không,....

Tìm lỗi không phải là trọng tâm chính trong kiểm thử chấp nhận, vì việc tìm lỗi đã được đội Developer và Tester thực hiện trong các giai đoạn kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống rồi.

Acceptance test gồm 2 loại kiểm thử là:

- **Alpha Test**, người dùng kiểm thử phần mềm ngay tại nơi phát triển phần mềm, lập trình viên sẽ ghi nhận các lỗi hoặc phản hồi, và lên kế hoạch sửa chữa.
- **Beta Test**, phần mềm sẽ được gửi tới cho người dùng để kiểm thử ngay trong môi trường thực, lỗi hoặc phản hồi cũng sẽ gửi ngược lại cho lập trình viên để sửa chữa. Lưu ý không nhất thiết phải thực hiện tất cả các loại kiểm tra nêu trên. Tùy yêu cầu và đặc trưng của từng hệ thống, tùy khả năng và thời gian cho phép của dự án, khi lập kế hoạch, trưởng dự án sẽ quyết định áp dụng những loại kiểm tra nào.