

# Chương 1

## TỔNG QUAN

### 1.1. Một số khái niệm

#### 1. Định nghĩa

**Bách khoa toàn thư mở Wikipedia:** Kiểm thử phần mềm là hoạt động khảo sát thực tiễn sản phẩm hay dịch vụ phần mềm trong đúng môi trường chúng dự định sẽ được triển khai nhằm cung cấp cho người có lợi ích liên quan những thông tin về chất lượng của sản phẩm hay dịch vụ phần mềm ấy. Mục đích của kiểm thử phần mềm là tìm ra các lỗi hay khiếm khuyết phần mềm nhằm đảm bảo hiệu quả hoạt động tối ưu của phần mềm trong nhiều ngành khác nhau.

**Glen Myers:** Việc kiểm thử là quá trình thực thi một chương trình với mục đích là tìm ra lỗi.

#### 2. Thuật ngữ

- **Lỗi (Error):** Là các lỗi lầm do con người gây ra.
- **Sai sót (Fault):** Sai sót gây ra lỗi. Có thể phân loại như sau:

Sai sót do đưa ra dư thừa - chúng ta đưa một vài thứ không chính xác vào mô tả yêu cầu phần mềm.

Sai sót do bỏ sót - Người thiết kế có thể gây ra sai sót do bỏ sót, kết quả là thiếu một số phần đáng ra phải có trong mô tả yêu cầu phần mềm.

- **Hỏng hóc (Failure):** Xảy ra khi sai sót được thực thi. (Khi thực thi chương trình tại các nơi bị sai thì sẽ xảy ra trạng thái hỏng hóc).

- **Kết quả không mong đợi, hậu quả (Incident):** Là những kết quả do sai sót đem đến. Hậu quả là các triệu chứng liên kết với một hỏng hóc và báo hiệu cho người dùng biết sự xuất hiện của hỏng hóc.

- **Trường hợp thử** (Test case): Trường hợp thử được liên kết tương ứng với hoạt động của chương trình. Một trường hợp thử bao gồm một tập các giá trị đầu vào và một danh sách các kết quả đầu ra mong muốn.

- **Thẩm tra** (Verification): Thẩm tra là tiến trình nhằm xác định đầu ra của một công đoạn trong việc phát triển phần mềm phù hợp với công đoạn trước đó.

=> Thẩm tra quan tâm đến việc ngăn chặn lỗi giữa các công đoạn

- **Xác nhận** (Validation): Xác nhận là tiến trình nhằm chỉ ra toàn hệ thống đã phát triển xong phù hợp với tài liệu mô tả yêu cầu.

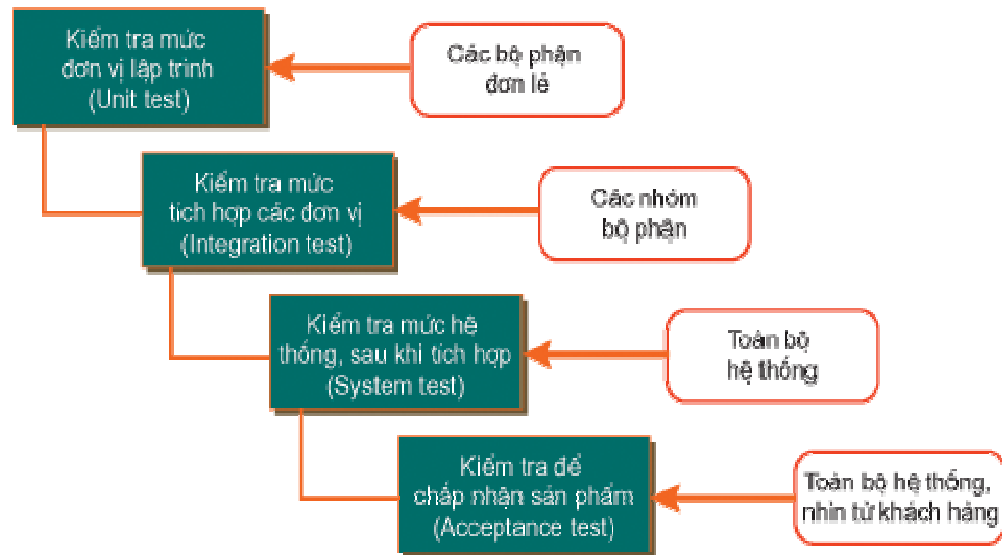
=> Xác nhận quan tâm đến sản phẩm cuối cùng không còn lỗi.

## **1.2. Giới thiệu về kiểm tra phần mềm và vòng đời của việc kiểm thử phần mềm**

### **1. Giới thiệu về kiểm tra phần mềm**

Một hoạt động mang tính sống còn trong các dự án sản xuất hoặc gia công phần mềm đó là kiểm thử. Người làm phần mềm chắc hẳn không ai nghi ngờ vai trò quan trọng của nó, tuy nhiên không phải ai (cả trong ngành và ngoài ngành) cũng hiểu rõ hoạt động này. Bản thân công việc kiểm tra phần mềm cũng là một lĩnh vực hoạt động độc lập và khá "hấp dẫn". Cùng với các dự án gia công sản xuất PM, hiện cũng có khá nhiều dự án mà nội dung công việc chỉ là kiểm tra những phần mềm đã được khách hàng phát triển sẵn. Mặc dù công việc kiểm tra phần mềm không xa lạ song những khái niệm và kỹ thuật lại khá rắc rối. Ở đây, chúng ta cần một cái nhìn tương đối bao quát về lĩnh vực "tưởng cũ nhưng không cũ" này.

Kiểm tra phần mềm đứng ở vị trí hết sức nhạy cảm, nó là bước đệm giữa giai đoạn xây dựng phần mềm và sử dụng phần mềm, trước khi giao sản phẩm hoàn chỉnh cho khách hàng. Mức độ cơ bản của kiểm tra phần mềm có thể nhận thấy như sau:



## Nhiệm vụ của kiểm tra phần mềm

Kiểm tra phần mềm phải thực hiện hai nhiệm vụ: xác minh và thẩm định.

Xác minh và thẩm định một hệ phần mềm là một quá trình liên tục xuyên suốt mọi giai đoạn của quá trình phần mềm. Xác minh và thẩm định là từ chung cho các quá trình kiểm tra để đảm bảo rằng phần mềm thỏa mãn các yêu cầu của chúng và các yêu cầu đó thỏa mãn các nhu cầu của người sử dụng phần mềm.

Xác minh và thẩm định là một quá trình kéo dài suốt vòng đời. Nó bắt đầu khi duyệt xét yêu cầu. Xác minh và thẩm định có hai mục tiêu:

- i) Phát hiện các khuyết tật trong hệ thống.
- ii) Đánh giá xem hệ thống liệu có dùng được hay không?

Sự khác nhau giữa xác minh và thẩm định là:

- i) Thẩm định: Xem xét cái được xây dựng có *là sản phẩm đúng* không? Tức là kiểm tra xem chương trình có được như mong đợi của người dùng hay không.
- ii) Xác minh: Xem xét cái được xây dựng có *đúng là sản phẩm* không? Như thế, xác minh là kiểm tra chương trình có phù hợp với đặc tả hay không.

Như trên, kiểm tra là một quá trình của tìm kiếm lỗi. Một kiểm tra tốt có khả năng cao tìm kiếm các lỗi chưa được phát hiện. Một kiểm tra thành công là kiểm tra tìm ra các lỗi mới, một kiểm tra tồi là kiểm tra mà không tìm được lỗi.

Có hai kiểu lỗi trong ứng dụng và các kiểm tra tốt sẽ xác định cả hai loại đó. Cụ thể:

- Không làm những điều cần phải làm: lỗi bỏ sót thường xuất hiện đối với ứng dụng mới được phát triển.
- Làm những điều không cần làm: lỗi của lệnh đã cư trú trước trong các ứng dụng bảo trì.

Các kiểm tra có mặt tại các mức khác nhau và được tiến hành bởi các cá nhân khác nhau trong quá trình phát triển ứng dụng. Các kiểm tra được tiến hành bởi đội ngũ dự án và kiểm tra được tiến hành bởi các cơ quan bên ngoài để chấp nhận ứng dụng.

Các kiểm tra của đội dự án được gọi là kiểm tra phát triển (Development test). Như đã nói trên, kiểm tra phát triển bao gồm kiểm tra đơn vị, kiểm tra hệ thống con, kiểm tra tích hợp và các kiểm tra hệ thống.

Các kiểm tra bởi các cơ quan bên ngoài được gọi là đảm bảo chất lượng (Quality assurance-QA) và kiểm tra chấp nhận (Acceptance test). Người ngoài có thể là người sử dụng hoặc đại diện người dùng, nó tạo một mục đích, đánh giá khách quan về ứng dụng và cơ quan bên ngoài được coi là có mục đích hơn các thành viên nhóm.

Kiểm tra đảm bảo chất lượng tương tự các kiểm tra hệ thống về mặt mục đích và tiến hành, nhưng nó khác là họ nằm ngoài sự điều khiển của dự án trưởng. Các báo cáo kiểm tra đảm bảo chất lượng được gửi thường xuyên tới nhóm phát triển và quản lý dự án. Các kiểm tra viên đảm bảo chất lượng lập kế hoạch cho chiến lược của mình và tiến hành kiểm tra để đảm bảo các ứng dụng thực hiện tất cả các chức năng cần thiết. Kiểm tra đảm bảo chất lượng là kiểm tra cuối cùng được làm trước khi ứng dụng được đưa sản xuất đại trà.

## **Mục tiêu kiểm thử**

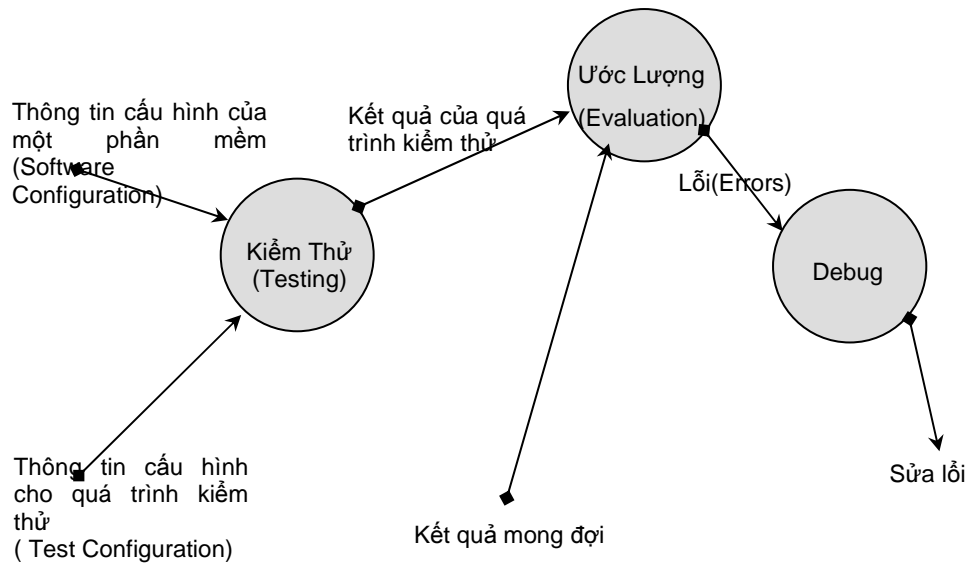
- Kiểm thử là một quá trình thực thi chương trình với mục đích là tìm ra lỗi/các yếu điểm của chương trình.
- Một trường hợp kiểm thử tốt là một trường hợp có khả năng lớn trong việc tìm ra những lỗi chưa được phát hiện.
- Một trường hợp kiểm không tốt (không thành công) là một trường hợp mà khả năng tìm thấy những lỗi chưa biết đến là rất ít.

Mục tiêu của kiểm thử phần mềm là thiết kế những trường hợp kiểm thử để có thể phát hiện một cách có thể thống những loại lỗi khác nhau và thực hiện việc đó với lượng thời gian và tài nguyên ít nhất có thể.

## **Quá trình kiểm thử**

Một mô hình cho quá trình kiểm thử được mô tả như hình dưới đây. Thông tin đầu vào được cung cấp cho quá trình kiểm thử gồm :

- Thông tin cấu hình của phần mềm: các thông tin này bao gồm: mô tả về yêu cầu của phần mềm (Software Requirement Specification). Mô tả về thiết kế của chương trình (Design Specification) và mã của chương trình.
- Thông tin cấu hình về kiểm thử bao gồm : kế hoạch kiểm thử, và thủ tục kiểm thử và các chương trình chạy kiểm thử như: chương trình giả lập môi trường, chương trình tạo các trường hợp kiểm thử... Các trường hợp kiểm thử phải đi cùng với kết quả mong muốn, Trong thực tế những thông tin này cũng là một phần của Thông tin cấu hình của một phần mềm ở trên.



Từ những thông tin đầu vào chương trình được chạy kiểm thử và kết quả của sau bước này sẽ được đánh giá/so sánh với một tập các kết quả mong đợi. khi kết quả của quá trình so sánh thất bại thì một lỗi được phát hiện và quá trình gỡ lỗi (debugging) bắt đầu. Gỡ lỗi là một quá trình không thể đoán trước được do một lỗi gây ra bởi sự khác nhau giữa kết quả kiểm thử và kết quả mong đợi có thể tốn một giờ, một ngày hay một tháng để tìm ra nguyên nhân và chỉnh sửa. Và cũng chính sự không chắc chắn cố hữu này mà làm cho quá trình kiểm định rất khó đưa ra một lịch biểu chắc chắn.

Lúc mà kết quả kiểm thử được thống kê và đánh giá. chất lượng và độ tin cậy của một phần mềm được ước lượng. Nếu có những lỗi nghiêm trọng xảy ra thường xuyên những lỗi dẫn đến cần phải thay đổi thiết kế của chương trình thì chất lượng của chương trình rất không tốt. Nhưng nếu ngược lại các module/hàm đều hoạt động đúng đắn như thiết kế ban đầu và những lỗi được tìm thấy có thể chỉnh sửa dễ dàng, thì có 2 kết luận có thể được đưa ra :

- Chất lượng của phần mềm là chấp nhận được
- Những kiểm định có thể không thoả đáng/thích hợp để phát hiện ra những lỗi nghiêm trọng đã đề cập trên.

Vậy thì cuối cùng là nếu mà quá trình kiểm thử phát hiện không có lỗi thì. Ở đây có một chút nghi ngờ rằng những thông tin cấu hình về kiểm thử không đủ và rằng lỗi vẫn tồn tại trong

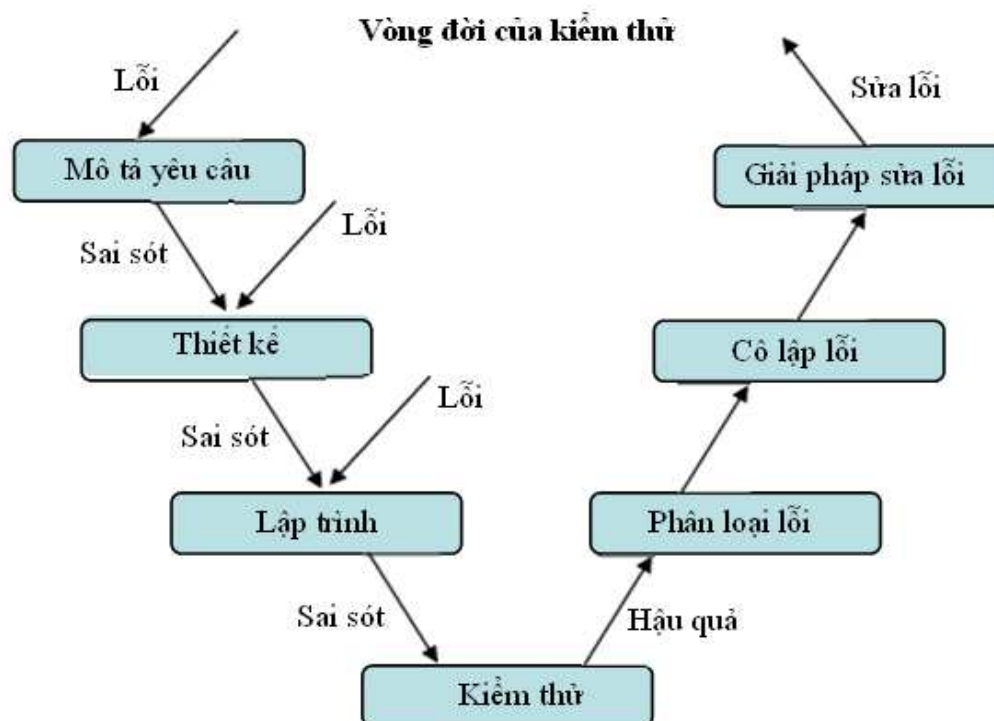
phần mềm. Những lỗi này sẽ được phát hiện sau này bởi người sử dụng và được chỉnh sửa bởi lập trình viên nhưng ở tại giai đoạn bảo trì và chi phí của những công việc này sẽ tăng lên 60 đến 100 lần so với chi phí cho mỗi chỉnh sửa trong giai đoạn phát triển.

Ta thấy rằng chi phí tiêu tốn quá nhiều cho quá trình bảo trì để chỉnh sửa một lỗi do đó cần phải có những kỹ thuật hiệu quả để tạo được các trường hợp kiểm thử tốt.

## 2. Vòng đời của việc kiểm thử phần mềm

Vòng đời của việc kiểm thử được thể hiện ở hình sau. Lỗi có thể xảy ra trong tất cả các công đoạn từ “Mô tả yêu cầu”, “Thiết kế” đến “Lập trình”. Từ công đoạn này chuyển sang công đoạn khác thường nảy sinh các sai sót (do dư thừa hoặc thiếu theo mô tả yêu cầu).

=> Công đoạn kiểm thử chúng ta sẽ phát hiện ra các hậu quả (các kết quả không mong muốn). Quá trình sửa lỗi bao gồm “phân loại lỗi”, “cô lập lỗi” (tìm ra nguyên nhân và nơi gây lỗi), đề ra “giải pháp sửa lỗi” và cuối cùng là khắc phục lỗi.



### 1.3. Các loại kiểm thử và phân loại

Có 2 mức phân loại kiểm thử: phân biệt theo mức độ chi tiết và phân biệt theo phương

pháp thử nghiệm

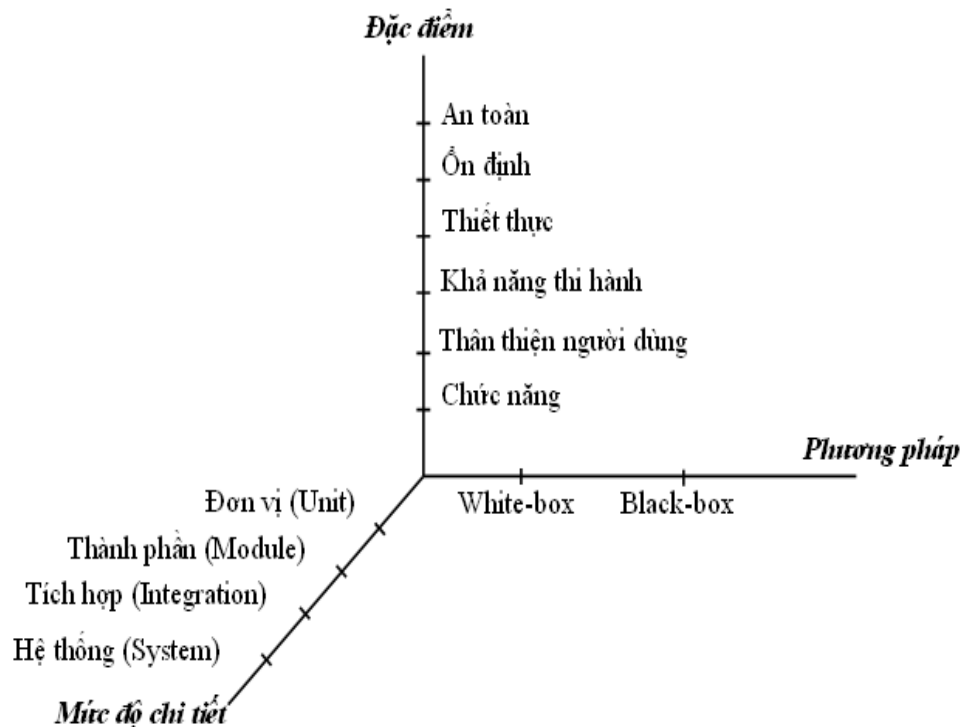
**Phân biệt theo mức độ chi tiết của các bộ phận hợp thành phần mềm.**

- Mức kiểm tra đơn vị (Unit)
- Mức kiểm tra hệ thống (System)
- Mức kiểm tra tích hợp (Integration)

**Phân biệt theo phương pháp thử nghiệm**

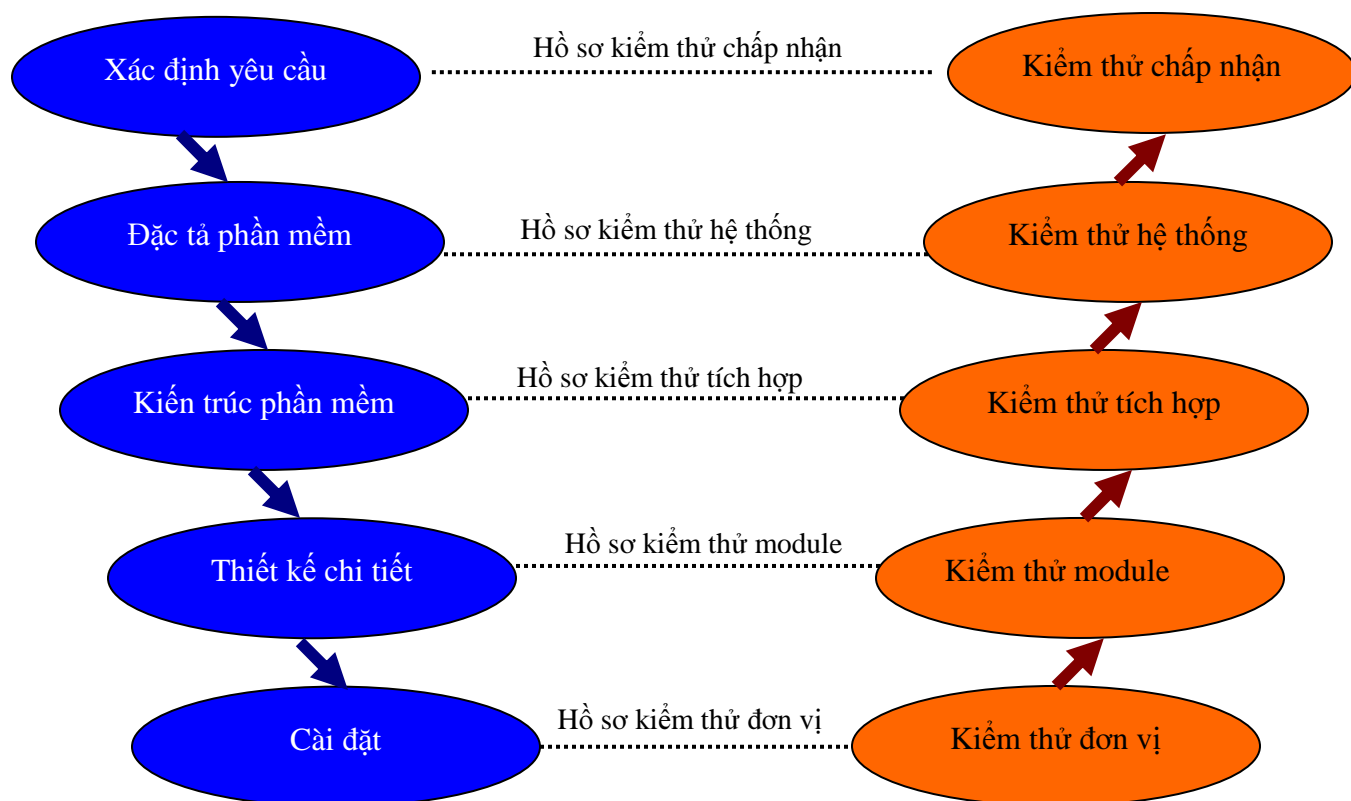
- Kiểm thử hộp đen (Black box testing) dùng để kiểm tra chức năng.
- Kiểm thử hộp trắng (White box testing) dùng để kiểm tra cấu trúc.

Hình bên dưới biểu diễn sự tương quan của “các tiêu chí chất lượng phần mềm”, “mức độ chi tiết đơn vị” và “phương pháp kiểm nghiệm”



Sự tương quan giữa các công đoạn xây dựng phần mềm và loại kiểm thử được thể hiện qua mô hình chữ V như sau:





#### 1.4. Sơ lược các kỹ thuật và phân loại kiểm thử phần mềm

Các kỹ thuật và công đoạn kiểm thử có thể chia như sau: Kiểm thử tầm hẹp và Kiểm thử tầm rộng.

**1. Kiểm thử tầm hẹp:** kiểm thử các bộ phận riêng rẽ.

*a. Kiểm thử hộp trắng (white-box testing)*

Còn gọi là kiểm thử cấu trúc. Kiểm thử theo cách này là loại kiểm thử sử dụng các thông tin về cấu trúc bên trong của ứng dụng. Việc kiểm thử này dựa trên quá trình thực hiện xây dựng phần mềm. Tiêu chuẩn của kiểm thử hộp trắng phải đáp ứng các yêu cầu như sau:

*Bao phủ dòng lệnh:* mỗi dòng lệnh ít nhất phải được thực thi 1 lần

*Bao phủ nhánh:* mỗi nhánh trong sơ đồ điều khiển (control graph) phải được đi qua một lần.

*Bao phủ đường:* tất cả các đường (path) từ điểm khởi tạo đến điểm cuối cùng trong sơ đồ dòng điều khiển phải được đi qua.

### *b. Kiểm thử hộp đen (black-box testing)*

Còn gọi là kiểm thử chức năng. Việc kiểm thử này được thực hiện mà không cần quan tâm đến các thiết kế và viết mã của chương trình. Kiểm thử theo cách này chỉ quan tâm đến chức năng đã đề ra của chương trình. Vì vậy kiểm thử loại này chỉ dựa vào bản mô tả chức năng của chương trình, xem chương trình có thực sự cung cấp đúng chức năng đã mô tả trong bản chức năng hay không mà thôi.

Kiểm thử hộp đen dựa vào các định nghĩa về chức năng của chương trình. Các trường hợp thử nghiệm (test case) sẽ được tạo ra dựa nhiều vào bản mô tả chức năng chứ không phải dựa vào cấu trúc của chương trình.

### *c. Vấn đề kiểm thử tại biên:*

Kiểm thử biên (boundary) là vấn đề được đặt ra trong cả hai loại kiểm thử hộp đen và hộp trắng. Lý do là do lỗi thường xảy ra tại vùng này.

## **2. Kiểm thử tầm rộng:**

- Kiểm thử bộ phận (Module testing): kiểm tra một bộ phận riêng rẽ.
- Kiểm thử tích hợp (Integration testing): tích hợp các bộ phận và hệ thống con.
- Kiểm thử hệ thống (System testing): kiểm thử toàn bộ hệ thống.

Mục đích của kiểm thử hệ thống là để đảm bảo toàn bộ hệ thống hoạt động như ý mà khách hàng mong muốn.

Bao gồm các loại kiểm thử sau:

- + Kiểm thử chức năng (Function testing)
- + Kiểm tra hệ thống sau khi tích hợp có hoạt động đúng chức năng với yêu cầu đặt ra trong bản mô tả yêu cầu hay không.

Ví dụ: với hệ thống xử lý văn bản thì kiểm tra các chức năng *tạo tài liệu, sửa tài liệu, xóa tài liệu...* có hoạt động hay không.

- + Kiểm thử hiệu suất (Performance testing)

+ Kiểm thử mức độ đáp ứng (stress testing): Thực thi hệ thống với giả thiết là các tài nguyên hệ thống yêu cầu không đáp ứng được về chất lượng, ổn định và số lượng.

+ Kiểm thử cấu hình (configuration tessting)

+ Phân tích hệ thống với các thiết lập cấu hình khác nhau.

+ Kiểm thử ổn định (robustness tessting): Kiểm thử dưới các điều kiện không mong đợi ví dụ như người dùng gõ lệnh sai, nguồn điện bị ngắt.

+ Kiểm thử hồi phục (recovery testing): Chỉ ra các kết quả trả về khi xảy ra lỗi, mất dữ liệu, thiết bị, dịch vụ... hoặc xóa các dữ liệu hệ thống và xem khả năng phục hồi của nó.

+ Kiểm thử quá tải (overload testing): Đánh giá hệ thống khi nó vượt qua giới hạn cho phép.

Ví dụ: Một hệ thống giao tác được yêu cầu thực thi 20 giao tác/giây. Khi đó sẽ kiểm tra nếu 30 giao tác/giây thì như thế nào?

+ Kiểm thử chất lượng (quality testing): Đánh giá sự tin tưởng, vấn đề duy tu, tính sẵn sàng của hệ thống. Bao gồm cả việc tính toán thời gian trung bình hệ thống sẽ bị hỏng và thời gian trung bình để khắc phục.

+ Kiểm thử cài đặt (Installation testing): Người dùng sử dụng các chức năng của hệ thống và ghi lại các lỗi tại vị trí sử dụng thật sự.

– Kiểm thử chấp nhận (Acceptance testing): thực hiện bởi khách hàng. Nhằm đảm bảo việc người dùng có được hệ thống mà họ yêu cầu. Việc kiểm thử này hoàn thành bởi người dùng phụ thuộc vào các hiểu biết của họ vào các yêu cầu.