

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN**



# **BÁO CÁO THỰC TẬP DOANH NGHIỆP**

**Đề tài: Test**

<b>Họ và tên sinh viên thực tập</b>	<b>: Hoàng Thạch Hùng</b>
<b>Mã số sinh viên</b>	<b>: B20DCAT074</b>
<b>Công ty thực tập</b>	<b>: VCCorp</b>
<b>Người phụ trách</b>	<b>: Anh Nguyễn Đình Lượng</b>
<b>Giảng viên hướng dẫn</b>	<b>: Cô Ninh Thị Thu Trang</b>

**Hà Nội – 6/2024**

## A. Đề bài

**Đề bài:** Thực hiện quy trình test dự án mới theo phạm vi:

- 1/ Test giao diện
- 2/ Test chức năng
- 3/ Test tối ưu SEO (tối ưu SEO onpage)
- 4/ Test page speed (tốc độ tải trang dựa theo công cụ PageSpeed Insights của Google )
- 4/ Test bảo mật (các lỗi bảo mật cơ bản)

Đây là một website cũ, được VCCorp code lại với code mới và giao diện mới trải nghiệm tốt hơn, bảo mật hơn, tối ưu SEO hơn..

Link website cũ: <https://thanhnienviet.vn/>

Link thiết kế mới:

<https://www.figma.com/design/t7kUEodRciAeDDjd0d9SUn/Biz-x-TapchiTNV?node-id=0-1&t=3i4FP2m4q44sF0V6-0>

**Công việc cần làm:**

- 1/ Phân tích dự án và Lên plan test
- 2/ Viết test case
- 3/ Thực thi test
- 4/ Báo cáo kết quả test

Thời gian 2 tuần kể từ nay 24/6 tới 6/7

**Thanhnienviet.vn - Tạp chí điện tử của TRUNG...**

**Tạp chí điện tử của TRUNG ƯƠNG ĐOÀN TNCS  
HỒ CHÍ MINH**

Trang thông tin chính thức của trung ương đoàn  
thanh niên cộng sản Hồ Chí Minh, cập nhật tin t...



11:02 AM

## B. Lí thuyết

### I. Các khái niệm cơ bản về kiểm thử phần mềm

#### 1. Kiểm thử phần mềm là gì?

Kiểm thử phần mềm là hoạt động nhằm tìm kiếm, phát hiện các lỗi của phần mềm; đảm bảo sản phẩm phần mềm đáp ứng chính xác, đầy đủ và đúng theo yêu cầu của khách hàng đã đặt ra; cung cấp mục tiêu, cái nhìn độc lập về phần mềm, cho phép việc đánh giá và hiểu rõ hơn các rủi ro khi thực thi phần mềm, tạo điều kiện cho bạn tận dụng tối đa tư duy đánh giá và sáng tạo để bạn có thể phát hiện ra những điểm mà người khác chưa tìm thấy.

#### 2. Lỗi phần mềm là gì? Nguyên nhân sinh ra lỗi của phần mềm.

Lỗi phần mềm: là một thất bại hoặc sai sót gây ra kết quả sai hoặc không mong muốn trong một chương trình. Đó là một lỗi khiến cho ứng dụng không thể vận hành như mong muốn.

Nguyên nhân sinh ra lỗi của phần mềm: Có rất nhiều nguyên nhân dẫn đến lỗi phần mềm. Lý do thường gặp nhất đó là do sai sót của con người trong quá trình thiết kế và lập trình. Khi đã biết được nguyên nhân dẫn đến những khiếm khuyết của phần mềm, việc sửa chữa để giảm thiểu những khiếm khuyết đó sẽ trở nên dễ dàng hơn rất nhiều.

Các lỗi phổ biến như:

- Hiểu nhầm trong giao tiếp hoặc không có giao tiếp.
- Sự phức tạp của các ứng dụng phần mềm hiện hành
- Lỗi lập trình
- Thay đổi yêu cầu
- Áp lực thời gian
- Các công cụ phát triển phần mềm
- Viết code hoặc kiểm thử các trường hợp khi chưa hiểu rõ tất cả các yêu cầu
- Thiết kế sai dẫn đến việc trong tất cả các giai đoạn của chu kỳ phát triển phần mềm đều có vấn đề.
- Dành ít thời gian hoặc bỏ qua hoàn toàn việc kiểm tra hồi quy.
- Không theo dõi quá trình phát triển và kiểm thử một cách liên mạch.

### **3. Vai trò của kiểm thử phần mềm**

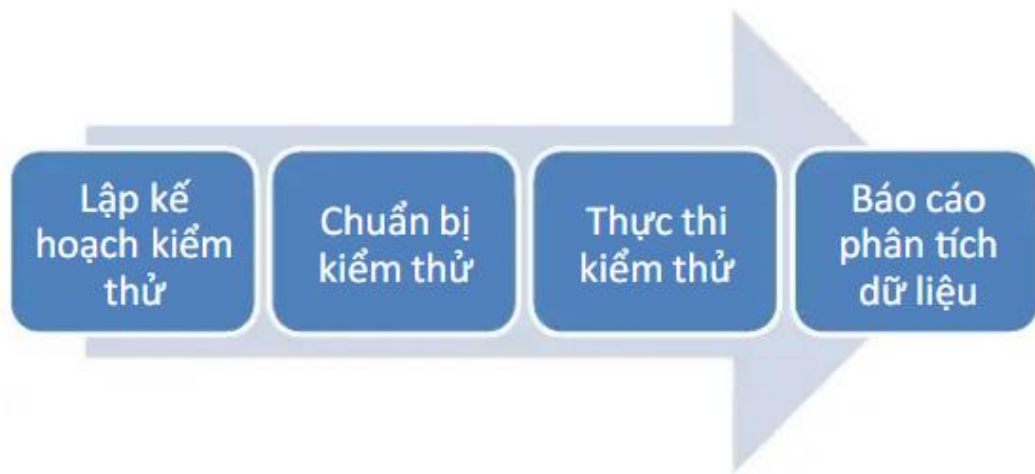
Lỗi có thể xuất hiện ở bất kì giai đoạn nào trong vòng đời phát triển phần mềm, kiểm tra nghiêm ngặt là cần thiết trong quá trình phát triển và bảo trì để xác định các lỗi để giảm thất bại khi hoạt động và làm tăng chất lượng của hệ thống khi đi vào vận hành.

- Tìm các bug (lỗi) phát sinh do lập trình viên tạo ra khi code.
- Đạt được sự tự tin và cung cấp thông tin về mức độ chất lượng.
- Để ngăn ngừa lỗi.
- Đảm bảo rằng kết quả cuối cùng đáp ứng các yêu cầu kinh doanh và người sử dụng.
- Để đạt được sự tin nhiệm của khách hàng bằng cách cung cấp cho họ một sản phẩm chất lượng.

Kiểm thử phần mềm sẽ giúp hoàn thiện các ứng dụng phần mềm hoặc sản phẩm so với yêu cầu kinh doanh và người sử dụng. Nó rất quan trọng để đảm bảo

kiểm thử tốt, kiểm thử các ứng dụng phần mềm hoàn toàn và chắc chắn rằng nó hoạt động tốt.

## II. Quy trình kiểm thử phần mềm



### 1. Lập kế hoạch kiểm thử (Test plan)

Nhiệm vụ quan trọng là xác định được các yếu tố sau:

- Các giai đoạn kiểm thử áp dụng cho dự án
- Các phương pháp kiểm thử
- Các công cụ kiểm thử
- Nhân sự tham gia
- Tài nguyên sử dụng, bao gồm các phần cứng và phần mềm
- Kế hoạch thực hiện
- Mốc bàn giao các tài liệu kiểm thử

### 2. Chuẩn bị kiểm thử

- Tìm hiểu nghiệp vụ của hệ thống phải kiểm thử
- Xây dựng kịch bản kiểm thử, phát triển các thủ tục và các kịch bản kiểm thử tự động (nếu có)
- Chuẩn bị dữ liệu kiểm thử
- Xem xét phê duyệt các tài liệu kiểm thử

### 3. Thực hiện kiểm thử

- Thực hiện test dựa trên testcase đã viết.
- Chạy lại các case bị lỗi trước đó để xác nhận là case đó đã được sửa.
- So sánh kết quả ghi nhận được khi thực thi với kết quả mong đợi.
- Đánh giá kết quả kiểm thử cho các trường hợp kiểm thử.

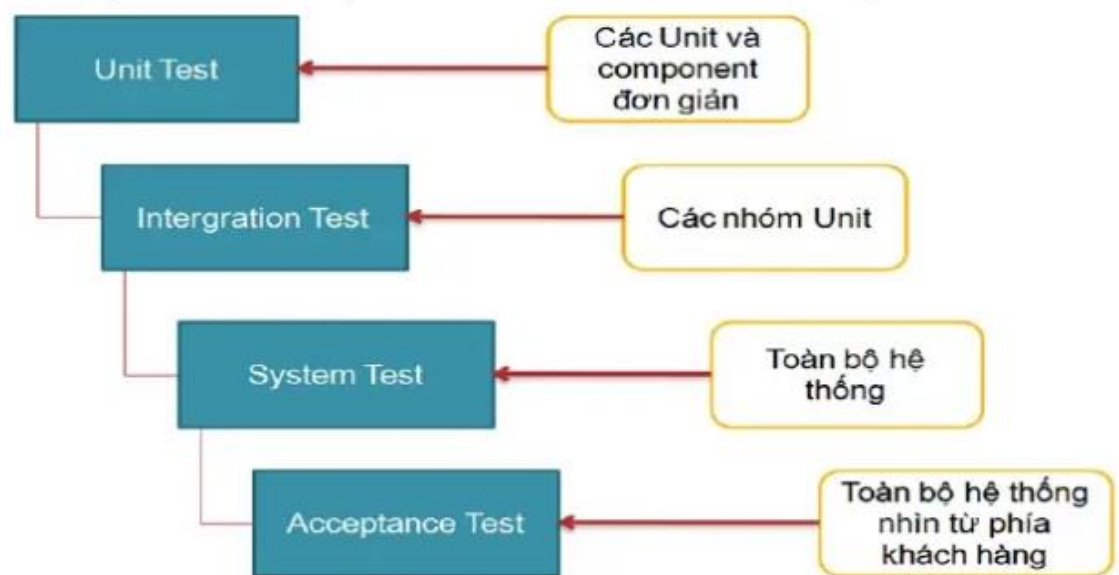
#### 4. Viết báo cáo kiểm thử (Test report)

Báo cáo kiểm thử thể hiện tiến độ kiểm thử, tiến độ sửa lỗi và số lượng lỗi được tìm thấy hay còn tồn của dự án.

Báo cáo là công cụ để phục vụ cho đánh giá hay giám sát dự án có kịp tiến độ hay không, có thể bàn giao cho khách hàng hay không và các vấn đề cần giải quyết khi mà số lượng lỗi còn nhiều, gây ra các rủi ro về tiến độ hoàn thành của dự án để có những điều chỉnh kịp thời.

### III. Các mức kiểm thử (Test Level)

Quy trình kiểm thử phần mềm sẽ được thực hiện theo 4 giai đoạn như sau:



#### 1. Kiểm thử đơn vị (Unit test)

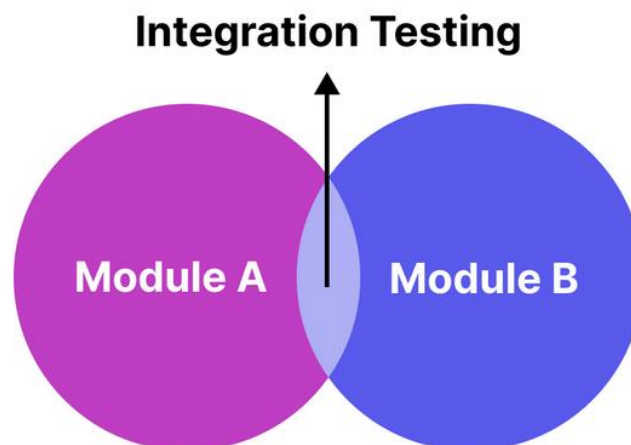


**Đơn vị:** Là thành phần nhỏ nhất của phần mềm có thể kiểm thử được. Ví dụ: Các hàm, lớp, thủ tục, phương thức. Đơn vị thường có kích thước nhỏ, chức năng hoạt động đơn giản, không gây nhiều khó khăn trong việc kiểm thử, ghi nhận và phân tích kết quả do đó nếu phát hiện lỗi việc tìm kiếm nguyên nhân và sửa lỗi cũng đơn giản và tốn ít chi phí hơn. Một nguyên lý đúc kết từ thực tiễn là thời gian dành cho kiểm thử đơn vị sẽ được đền bù bằng việc tiết kiệm được khá nhiều thời gian và chi phí cho việc kiểm thử và sửa lỗi ở các mức độ kiểm thử sau đó.

**Mục đích:** Đảm bảo thông tin được xử lý đúng và có đầu ra chính xác trong mối tương quan giữa dữ liệu nhập và chức năng của đơn vị.

**Người thực hiện:** Do việc kiểm thử đơn vị đòi hỏi phải kiểm tra từng nhánh lệnh, nên đòi hỏi người kiểm thử có kiến thức về lập trình cũng như về thiết kế của hệ thống nên người thực hiện thường là lập trình viên.

## **2. Kiểm thử tích hợp (Integration testing)**



Kiểm thử tích hợp là kiểm thử sự kết hợp và giao tiếp giữa các đơn vị của một chương trình và kiểm thử như một chương trình đã hoàn thành

**Mục đích:**

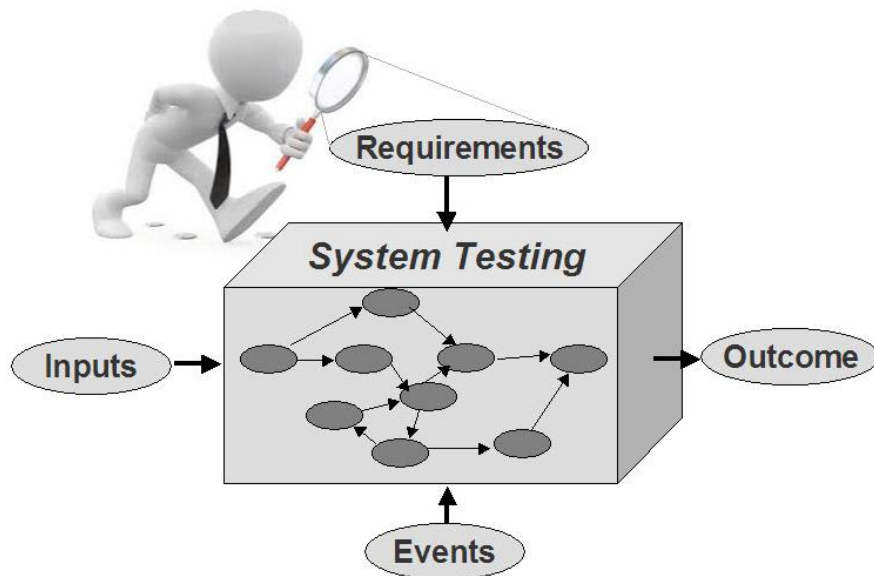
- Phát hiện lỗi giao tiếp xảy ra giữa các đơn vị cũng như lỗi của bản thân từng đơn vị (nếu có)
- Tích hợp các đơn vị lẻ thành các hệ thống nhỏ và cuối cùng là tích hợp các hệ thống nhỏ thành một hệ thống hoàn chỉnh để chuẩn bị cho kiểm thử hệ thống

**Người thực hiện:** thường là Lập trình viên

**Lưu ý:**

- Kiểm thử tích hợp chỉ nên thực hiện trên từng đơn vị đã được kiểm tra cẩn thận trước đó bằng kiểm thử đơn vị, và tất cả các lỗi mức đơn vị đã được sửa chữa
- Nên tích hợp dần từng đơn vị: Một đơn vị nên được tích hợp vào một nhóm các đơn vị khác đã được tích hợp và hoàn thành kiểm thử tích hợp trước đó, vì khi đó chỉ cần kiểm tra giao tiếp giữa đơn vị mới được thêm vào với nhóm các đơn vị đã được tích hợp trước đó

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



Kiểm thử hệ thống bắt đầu khi tất cả các đơn vị của hệ thống được tích hợp thành công, đây là công đoạn kiểm thử tốn nhiều công sức và thời gian nhất. Và đặc biệt, công đoạn này thường đòi hỏi được thực hiện bởi 1 nhóm nhân viên tách biệt với nhóm phát triển, có chuyên môn và kinh nghiệm kiểm thử.

Kiểm thử hệ thống gồm nhiều loại kiểm thử khác nhau, trong số đó, các mục tiêu kiểm thử quan trọng nhất là:

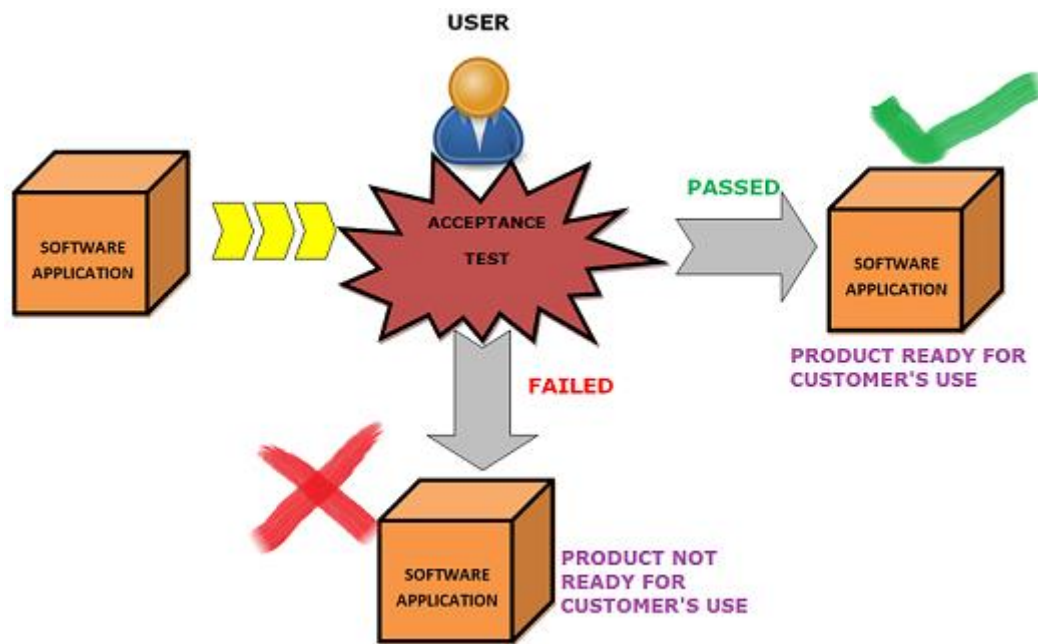
- Kiểm thử chức năng
- Kiểm thử hiệu năng
- Kiểm thử an toàn thông tin

Mục đích: Kiểm tra xem hệ thống được làm ra có thỏa mãn yêu cầu hay không về nhiều khía cạnh: hoạt động, độ tin cậy, hiệu năng của hệ thống

Người thực hiện: Nhóm nhân viên kiểm thử

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





Mục đích: Kiểm thử chấp nhận còn gọi là kiểm thử nghiệm thu nhằm mục đích chứng minh phần mềm thỏa mãn tất cả yêu cầu của khách hàng và khách hàng đã chấp nhận sản phẩm.

Người thực hiện: Khách hàng.

Có 2 loại kiểm thử chấp nhận đó là kiểm thử Alpha và kiểm thử Beta:

- Kiểm thử Alpha: Người dùng kiểm thử phần mềm ngay tại nơi phát triển phần mềm với sự hỗ trợ của nhân viên kiểm thử, nhân viên kiểm thử sẽ khi lại các lỗi hoặc phản hồi của khách hàng và báo lại với đơn vị phát triển phần mềm để lên kế hoạch sửa chữa.
- Kiểm thử Beta: là loại kiểm thử mà khách hàng thực hiện kiểm thử ở chính môi trường của họ. Loại kiểm thử này được thực hiện sau kiểm thử Alpha.

## 5. Kiểm thử hồi qui

Kiểm thử hồi qui là một hoạt động cần thiết để chỉ ra rằng việc thay đổi mã nguồn không gây ra những ảnh hưởng bất lợi đến hệ thống nói chung.

Mục đích kiểm thử: Kiểm thử hồi qui dùng để kiểm tra các phần được sửa chữa và các phần liên quan đến các phần sửa chữa trong phần mềm, để đảm bảo rằng những sự thay đổi đó không gây ra lỗi trong những phần khác.

Cách thực hiện: Tái sử dụng các kịch bản kiểm thử từ những phần kiểm thử trước để kiểm thử các module đã được sửa chữa. Sử dụng các công cụ kiểm thử tự động: Tạo một số test script về chức năng. Xây dựng một chương trình phân tích cơ sở hạ tầng. Chúng ta xây dựng một cơ sở hạ tầng có thể mở rộng được để



thực hiện và đánh giá chương trình phân tích. Dựa vào kết quả phân tích chúng ta xác định phạm vi cần kiểm thử hồi qui.

Điều kiện hoàn thành: Toàn bộ các trường hợp kiểm thử đã chọn được thực hiện và đạt yêu cầu.

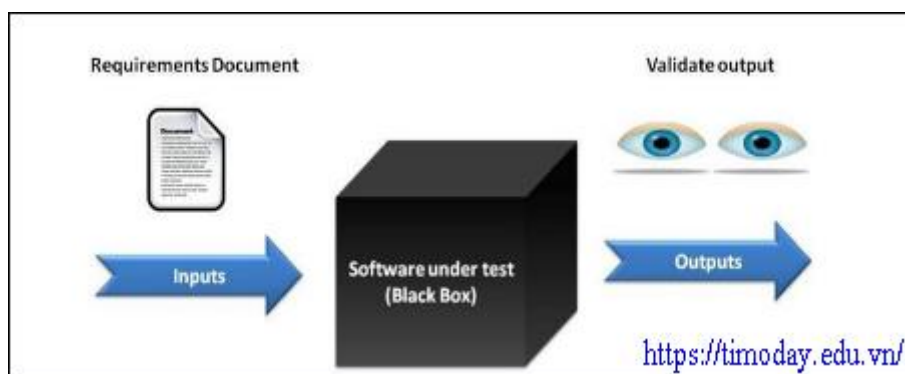
#### IV. Các kỹ thuật kiểm thử phần mềm

Có 3 kỹ thuật kiểm thử phần mềm chính là:

- Kiểm thử hộp đen
- Kiểm thử hộp trắng
- Kiểm thử hộp xám

##### 1. Phương pháp kiểm thử hộp đen (Black box testing)

**Khái niệm:** Là phương pháp kiểm thử dựa trên đầu vào và đầu ra của chương trình để kiểm thử. Chỉ kiểm thử chức năng và giao diện dựa trên nghiệp vụ của hệ thống mà không quan tâm tới mã chương trình bên trong được viết ra sao. Tester xem phần mềm như là một hộp đen. Kiểm thử hộp đen không yêu cầu kỹ sư kiểm thử cần phải có bất kỳ kiến thức về mã hoặc thuật toán của chương trình. Nó kiểm tra các chức năng của hệ thống tức là những gì hệ thống được cho là cần phải làm dựa trên các đặc tả yêu cầu (Requirement document).



##### Ưu điểm:

- Không có mối ràng buộc nào về code, và kiểm thử những thứ lập trình viên có thể bỏ qua hoặc không nhìn thấy trong quá trình lập trình.
- Người kiểm thử thực hiện từ quan điểm của người dùng và sẽ giúp đỡ trong việc sáng tỏ sự chênh lệch về thông số kỹ thuật.
- Người kiểm thử có thể không phải là một lập trình viên chuyên nghiệp, không cần phải biết ngôn ngữ lập trình hoặc làm thế nào các phần mềm đã được thực hiện.

- Người kiểm thử có thể thực hiện một cách độc lập từ các developer, cho phép một cái nhìn khách quan và tránh sự phát triển thiên vị.
- Thiết kế kịch bản kiểm thử khá nhanh, ngay khi mà các yêu cầu chức năng được xác định.

### **Nhược điểm:**

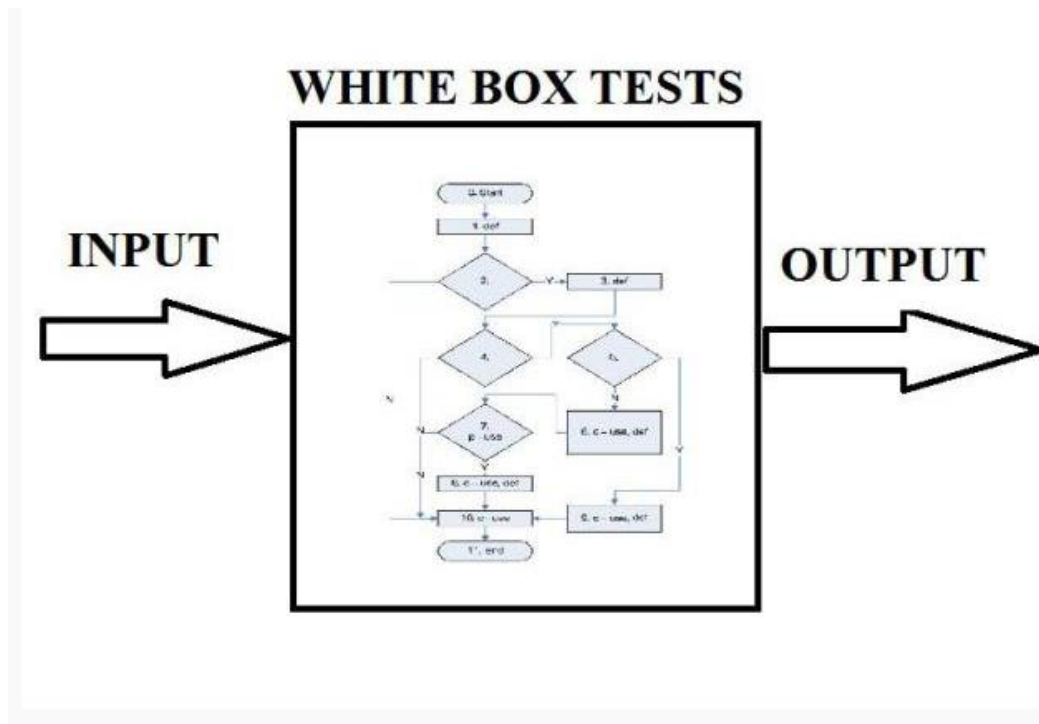
- Dữ liệu đầu vào yêu cầu một khối lượng mẫu khá lớn.
- Chỉ có thể khám phá mù (không biết phần mềm kiểm thử được xây dựng như thế nào), do đó khi áp dụng phương pháp kiểm thử hộp đen đòi hỏi người thực hiện phải làm việc vất vả hơn để khám phá được càng nhiều bug càng tốt.
- Nhiều dự án không có thông số rõ ràng thì việc thiết kế test case rất khó và do đó khó viết kịch bản kiểm thử do cần xác định tất cả các yếu tố đầu vào, và thiếu cả thời gian cho việc tập hợp này.
- Chỉ có một số nhỏ các đầu vào có thể được kiểm tra và nhiều đường dẫn chương trình sẽ được để lại chưa được kiểm tra.

## **2. Phương pháp kiểm thử hộp trắng (White box testing)**

**Khái niệm:** Là phương pháp kiểm thử dựa cả vào giải thuật, cấu trúc code bên trong phần mềm, việc kiểm thử được tiến hành dựa cả vào việc kiểm xem giải thuật, mã lệnh đã làm có đúng không.

Trong kiểm thử hộp trắng, cấu trúc mã hoặc giải thuật của chương trình được đưa vào xem xét, các trường hợp kiểm thử được thiết kế dựa vào cấu trúc mã hoặc cách thức làm việc của chương trình. Người kiểm thử truy cập vào mã chương trình và có thể kiểm tra nó, lấy đó làm cơ sở để hỗ trợ việc kiểm thử.

Để thực hiện được phương pháp kiểm thử hộp trắng thì người kiểm thử phải có kỹ năng, kiến thức nhất định về ngôn ngữ lập trình được dùng, về thuật giải được dùng trong thành phần phần mềm để có thể thông hiểu được chi tiết về các đoạn code cần kiểm thử.



### Ưu điểm:

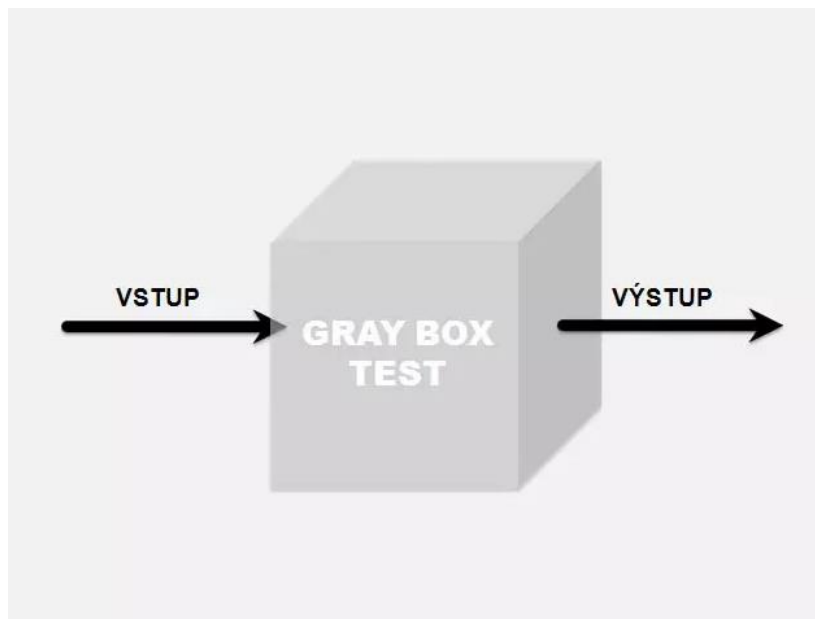
- Test có thể bắt đầu ở giai đoạn sớm hơn, không cần phải chờ đợi cho GUI để có thể test.
- Test kỹ càng hơn, có thể bao phủ hầu hết các trường hợp.
- Thích hợp trong việc tìm kiếm lỗi và các vấn đề trong mã lệnh.
- Cho phép tìm kiếm các lỗi ẩn bên trong.
- Các lập trình viên có thể tự kiểm tra.
- Giúp tối ưu việc mã hóa.
- Do yêu cầu kiến thức cấu trúc bên trong của phần mềm nên việc kiểm soát lỗi tối đa nhất.

### Nhược điểm:

- Yêu cầu người kiểm thử phải am hiểu cấu trúc mã lệnh của chương trình. Do đó đòi hỏi tài nguyên nhân lực và máy tính kém
- Có khả năng tồn tại các tổ hợp lệnh khác nhau gây lỗi
- Không kiểm thử hết đường đi với các vòng lặp
- Khó thực hiện và chi phí thực hiện cao

### 3. Phương pháp kiểm thử hộp xám

**Khái niệm:** là một phương pháp kiểm thử phần mềm được kết hợp giữa phương pháp kiểm thử hộp trắng và phương pháp kiểm thử hộp đen.



Trong kiểm thử hộp xám, cấu trúc bên trong sản phẩm chỉ được biết một phần, Tester có thể truy cập vào cấu trúc dữ liệu bên trong và thuật toán của chương trình với mục đích là để thiết kế testcase, nhưng khi test thì test như là người dùng cuối hoặc là ở mức hộp đen.

Được gọi là kiểm thử hộp xám vì trong chương trình phần mềm, mắt của Tester giống như hộp xám/bán trong suốt-nhìn qua hộp này ta chỉ có thể thấy được một phần.

Ví dụ: Khi code của 1 module nào đó được xem xét để thiết kế testcase (phương pháp kiểm thử hộp trắng) và khi test thực tế thì được thực hiện test trên giao diện người dùng (phương pháp kiểm thử hộp đen).

## **V. Các kỹ thuật kiểm thử**

Có nhiều kỹ thuật kiểm thử phần mềm, sau đây là 4 kỹ thuật phổ biến nhất trong kỹ thuật kiểm thử hộp đen là: kỹ thuật phân vùng tương đương, kỹ thuật phân tích giá trị biên, bảng quyết định và kỹ thuật đoán lỗi.

Trong quá trình kiểm thử, tester có thể áp dụng nhiều kỹ thuật khác nhau, kết hợp các phương pháp kiểm thử khác nhau để có thể tìm ra lỗi của phần mềm một cách tối đa.

### **1. Kỹ thuật phân vùng tương đương**

Đây là phương pháp chia đầu vào thành những nhóm tương đương nhau. Nếu một giá trị trong nhóm hoạt động đúng thì tất cả các giá trị trong nhóm đó cũng hoạt động đúng và ngược lại.

**Mục đích:** Giảm đáng kể số lượng test case cần phải thiết kế vì với mỗi lớp tương đương ta chỉ cần test trên các phần tử đại diện.

**Ưu điểm:** Vì mỗi vùng tương đương ta chỉ cần test trên các phần tử đại diện nên số lượng test case được giảm đi khá nhiều nhờ đó mà thời gian thực hiện test cũng giảm đáng kể.

**Nhược điểm:** Không phải với bất kỳ bài toán nào cũng có thể áp dụng kỹ thuật này. Có thể bị hack lỗi ở biên nếu chỉ chọn giá trị ở khoảng giữa của miền tương đương.

## 2. Bảng quyết định

Bảng quyết định là một đại diện bảng đầu vào gồm có các điều kiện đầu vào, các tình huống thử nghiệm và các kết quả. Đây là một công cụ rất hiệu quả được sử dụng cho cả kiểm thử phần mềm phức tạp và quản lý các yêu cầu.

Bảng quyết định là một kỹ thuật tốt để áp dụng cho những trường hợp cần nhiều sự kết hợp. Bảng quyết định hỗ trợ việc lựa chọn test case một cách có hệ thống và có thể đem lại nhiều lợi ích trong việc nhận biết vấn đề tiềm ẩn và sự không rõ ràng trong đặc tả (specification).

Bảng quyết định đảm bảo bao gồm tất cả các kết hợp có thể có của các giá trị điều kiện được gọi là thuộc tính tính đầy đủ, mang lại số testcase tối thiểu với độ bao phủ tối đa.

Phương pháp này khắc phục được khuyết điểm của 2 phương pháp phân tích giá trị biên và phân vùng tương đương, đó là kiểm soát được sự kết hợp của các giá trị đầu vào bằng cách sử dụng mô hình quan hệ logic nguyên nhân – kết quả cho các thành phần phần mềm.

## 3. Kỹ thuật phân tích giá trị biên

Đây là phương pháp kiểm thử mà chúng ta sẽ kiểm thử tất cả các giá trị ở vùng biên của dữ liệu vào và dữ liệu ra. Chúng ta sẽ tập trung vào các giá trị biên chứ không kiểm thử toàn bộ dữ liệu.

**Mục đích:** Thay vì chọn nhiều giá trị trong lớp đương tương để làm đại diện, phân tích giá trị biên yêu cầu chọn một hoặc vài giá trị là các cạnh của lớp tương đương để làm điều kiện kiểm thử.

Phân tích giá trị biên là kỹ thuật thiết kế test case và hoàn thành phân vùng tương đương.

Thiết kế test case bằng phương pháp phân tích giá trị biên chính là lựa chọn các test case để thực thi giá trị biên

Nguyên tắc:

- Giá trị nhỏ nhất
- Giá trị ngay dưới giá trị nhỏ nhất
- Giá trị bình thường
- Giá trị ngay trên giá trị lớn nhất
- Giá trị lớn nhất

#### 4. Kỹ thuật đoán lỗi

Phương pháp này không có quy trình cụ thể vì có tính trực giác cao và không thể dự đoán trước

Phương pháp dựa vào kinh nghiệm, trực quan của kiểm thử viên để đưa ra các trường hợp kiểm thử. Trong quá trình kiểm thử, kiểm thử viên có thể áp dụng nhiều kỹ thuật khác nhau, kết hợp các phương pháp kiểm thử với nhau để có thể tìm các lỗi của phần mềm một cách tối đa.

Ví dụ: Ở màn hình login, đôi khi dev code hay gán username là “Admin” và pass là rỗng hoặc “123”, vì vậy khi thực hiện test chúng ta nên test cả case này.

### VI. Phân loại kiểm thử

#### 1. Kiểm thử thủ công (Manual Test)

**Khái niệm:** Kiểm thử thủ công, Tester làm mọi công việc hoàn toàn bằng tay, từ viết test case đến thực hiện kiểm thử; mọi thao tác như: Nhập điều kiện đầu vào, thực hiện một số sự kiện khác nhau, sau đó so sánh kết quả thực tế với kết quả mong muốn trong test case, điền kết quả kiểm thử. Hiện nay, phần lớn các tổ chức, các công ty phần mềm, hoặc các nhóm làm phần mềm đều thực hiện kiểm thử thủ công là chủ yếu.

**Ưu điểm:**

- Cho phép Tester thực hiện việc kiểm thử khám phá.
- Thích hợp kiểm tra sản phẩm lần đầu tiên.
- Thích hợp kiểm thử trong trường hợp các test case chỉ thực hiện một số ít lần.
- Giảm được chi phí ngắn hạn.

**Nhược điểm:** Tốn thời gian. Đối với mỗi lần kiểm tra lại, người kiểm thử phải thực hiện lại một tập hợp các test case đã chạy, dẫn đến sự mệt mỏi và gây ra sự nhàm chán.

## **2. Kiểm thử tự động (Automation Test)**

**Khái niệm:** Kiểm thử phần mềm tự động là thực hiện kiểm thử phần mềm bằng một chương trình đặc biệt với rất ít hoặc không có sự tương tác của con người, giúp cho Tester không phải lặp đi lặp lại các bước nhàm chán.

### **Ưu điểm:**

- Thay thế con người lặp đi lặp lại đúng quy tắc các bước kiểm thử.
- Độ tin cậy cao: Dù lặp đi lặp lại nhiều lần vẫn cho ra kết quả giống nhau, do vậy độ ổn định cao, tránh được rủi ro có thể phát sinh.
- Cải thiện chất lượng: Kiểm thử tự động làm giảm rủi ro về mặt chất lượng sản phẩm, việc kiểm thử được thực hiện một cách nhanh chóng, không gây mệt mỏi, tránh trường hợp chủ quan cho những trường hợp kiểm thử hồi quy, retest, kiểm thử toàn bộ hệ thống; có thể tái sử dụng các trường hợp kiểm thử.
- Tốc độ xử lý cực nhanh.
- Chi phí thấp: Việc rút ngắn thời gian và tiết kiệm nhân lực giúp cho việc kiểm thử tự động trở nên hiệu quả.
- Hỗ trợ xuất ra nhiều ngôn ngữ phổ biến như Ruby, Java, Python...

### **Nhược điểm:**

- Ban đầu thì chi phí cho kiểm thử tự động sẽ cao hơn kiểm thử thủ công.
- Mất chi phí về bản quyền, bảo trì, tìm hiểu, training.
- Khó mở rộng hơn nhiều so với kiểm thử thủ công
- Yêu cầu những người có trình độ chuyên môn cao mới thực hiện được.
- Số lượng công việc phải làm để mở rộng kiểm thử tự động sẽ nhiều và khó hơn so với kiểm thử thủ công.