

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN
-----o0o-----

Thạc Bình Cường

Bài giảng điện tử môn học

**KIỂM THỬ VÀ BẢO ĐẢM
CHẤT LƯỢNG PHẦN MỀM**

MỞ ĐẦU.....	4
CHƯƠNG 1: CÁC KHÁI NIỆM	5
1.1. Các định nghĩa.....	5
1.2. Vòng đời của việc kiểm nghiệm (testing life cycle):.....	6
1.3. Phân loại kiểm nghiệm:	7
1.4. Sự tương quan giữa các công đoạn xây dựng phần mềm và loại kiểm nghiệm: Mô hình chữ V.....	8
1.5. Sơ lược các kỹ thuật và công đoạn kiểm nghiệm:.....	9
CHƯƠNG 2: KIỂM CHỨNG VÀ XÁC NHẬN (V & V)	13
2.1. Kiểm chứng và hợp lệ hoá.....	13
2.1.1. Tổ chức việc kiểm thử phần mềm	14
2.1.2. Chiến lược kiểm thử phần mềm	15
2.1.3. Tiêu chuẩn hoàn thành kiểm thử	17
2.2. Phát triển phần mềm phòng sạch (cleanroom software development).....	18
2.2.1. Nghệ thuật của việc gỡ rối.....	18
2.2.2. Tiến trình gỡ lỗi.....	18
2.2.3. Xem xét tâm lý	19
2.2.4. Cách tiếp cận gỡ lỗi	19
CHƯƠNG 3: KIỂM THỬ PHẦN MỀM.....	22
3.1. Quá trình kiểm thử.....	22
3.2. Kiểm thử hệ thống	24
3.3. Kiểm thử tích hợp.....	25
3.4. Kiểm thử phát hành	27
3.5. Kiểm thử hiệu năng	31
3.6. Kiểm thử thành phần	32
3.7. Kiểm thử giao diện	33
3.8. Thiết kế trường hợp thử (Test case design).....	35
3.9. Tự động hóa kiểm thử (Test automation).....	45
CHƯƠNG 4: CÁC PHƯƠNG PHÁP KIỂM THỬ	49
4.1. Phương pháp white-box:.....	50
4.2. Phương pháp black-box:.....	59
CHƯƠNG 5: KIỂM THỬ TÍCH HỢP.....	66
5.1. Tích hợp trên xuống.	66
5.2. Tích hợp dưới lên.	68
5.3. Kiểm thử nội quy.....	69
5.4. Gợi ý về việc kiểm thử tích hợp	71
5.5. Lập tài liệu về kiểm thử tích hợp.....	72
CHƯƠNG 6: KỸ NGHỆ ĐỘ TIN CẬY PHẦN MỀM	75
6.1. Giới thiệu.....	75
6.2. Xác nhận tính tin cậy	76
6.2.1. Sơ thảo hoạt động	78
6.2.2. Dự đoán tính tin cậy	79
6.3. Đảm bảo tính an toàn.....	82
6.3.1. Những luận chứng về tính an toàn.....	83
6.3.2. Đảm bảo quy trình	86
6.3.3. Kiểm tra tính an toàn khi thực hiện	88
6.4. Các trường hợp an toàn và tin cậy được.....	89

CHƯƠNG 7: KIỂM THỬ PHẦN MỀM TRONG CÔNG NGHIỆP	95
7.1. QUY TRÌNH KIỂM TRA PHẦN MỀM CƠ BẢN.....	95
7.2. MÔ HÌNH KIỂM TRA PHẦN MỀM TMM (TESTING MATURITY MODEL).....	99
7.3. Các công cụ kiểm thử (Test tools).....	105
7.3.1. TẠI SAO PHẢI DÙNG TEST TOOL	105
7.3.2. KHÁI QUÁT VỀ KTTĐ	106
7.3.3. GIỚI THIỆU CÔNG CỤ KTTĐ: QUICKTEST PROFESSIONAL.....	108
7.3.4. Kiểm thử đơn vị với JUnit	112
CHƯƠNG 8: ƯỚC LƯỢNG GIÁ THÀNH PHẦN MỀM.....	129
8.1. Giới thiệu	129
8.2. Năng suất phần mềm	131
8.3. Kỹ thuật ước lượng	135
8.4. Mô hình hoá chi phí thuật toán.....	137
8.5. Mô hình COCOMO	139
8.6. Mô hình chi phí giải thuật trong kế hoạch dự án.....	147
8.7. Nhân viên và khoảng thời gian của dự án	149
CHƯƠNG 9: QUẢN LÝ CHẤT LƯỢNG PHẦN MỀM	153
9.1. Chất lượng quá trình và chất lượng sản phẩm:.....	153
9.2. Chất lượng quá trình và chất lượng sản phẩm:.....	155
9.3. Đảm bảo chất lượng và các chuẩn chất lượng.....	156
9.4. Lập kế hoạch chất lượng.....	163
9.5. Kiểm soát chất lượng.....	164
9.6. CMM/CMMi	165
9.6.2. Cấu trúc của CMM	166
9.6.3. So sánh giữa CMM và CMMi	172
CHƯƠNG 10: QUẢN LÝ CẤU HÌNH.....	174
10.1. Giới thiệu	174
10.2. Kế hoạch quản trị cấu hình.....	176
11.2. Quản lý việc thay đổi.....	179
11.3. Quản lý phiên bản và bản phát hành.....	183
11.4. Quản lý bản phát hành.....	186
11.5. Xây dựng hệ thống	189
11.6. Các công cụ CASE cho quản trị cấu hình	190
PHỤ LỤC- CÁC CÂU HỎI ÔN TẬP.....	197
1. Chất lượng và đảm bảo chất lượng phần mềm.....	197
2. Các độ đo đặc trưng chất lượng phần mềm.....	198
3. Kiểm thử phần mềm	199
4. Quản lý cấu hình phần mềm.....	201
TÀI LIỆU THAM KHẢO.....	202

MỞ ĐẦU

Quản lý chất lượng phần mềm là vấn đề không mới nhưng theo một số đánh giá là còn yếu của các công ty phần mềm Việt Nam. Một số công ty trong nước hiện đã đạt các chuẩn quốc tế CMM/CMMI trong nâng cao năng lực và quản lý chất lượng phần mềm, song chỉ đếm được trên đầu ngón tay, và hiện cũng chỉ gói gọn trong vài công ty gia công cho thị trường nước ngoài.

Lâu nay, nói đến chất lượng phần mềm, không ít người nghĩ ngay đến vấn đề là xác định xem phần mềm đó có phát sinh lỗi hay không, có "chạy" đúng như yêu cầu hay không và cuối cùng thường quy về vai trò của hoạt động kiểm thử phần mềm (testing) như là hoạt động chịu trách nhiệm chính.

Với quan điểm của khách hàng, điều này có thể đúng, họ không cần quan tâm nội tình của hoạt động phát triển phần mềm, điều họ cần quan tâm là liệu sản phẩm cuối cùng giao cho họ có đúng hạn hay không và làm việc đúng như họ muốn hay không.

Tuy nhiên theo quan điểm của người phát triển phần mềm, thực tế cho thấy hoạt động kiểm thử phần mềm là quan trọng, nhưng không đủ để đảm bảo sản phẩm sẽ được hoàn thành đúng hạn và đúng yêu cầu. Kiểm thử sau cùng để phát hiện lỗi là điều tất nhiên phải làm, nhưng trong rất nhiều trường hợp, điều đó thường quá trễ và sẽ phải mất rất nhiều thời gian để sửa chữa.

Thực tế cho thấy, để đảm bảo được hai tiêu chí "đơn giản" trên của khách hàng, đòi hỏi tổ chức không chỉ vận hành tốt khâu kiểm thử phần mềm, mà phải tổ chức và duy trì sự hoạt động nhịp nhàng của cả một hệ thống các công việc liên quan đến một dự án phần mềm, từ đây xuất hiện một khái niệm có tên là "hệ thống quản lý chất lượng phần mềm" bao gồm các quy trình được thực thi xuyên suốt chu kỳ phát triển của dự án phần mềm song hành cùng việc kiểm thử phần mềm nhằm đảm bảo chất lượng cho phần mềm khi chuyển giao cho khách hàng.

Với thực tế trên, là những người làm công tác đào tạo mong muốn cung cấp cho sinh viên ngành công nghệ phần mềm - những người sẽ là nguồn nhân lực chủ yếu trong tương lai của các doanh nghiệp phần mềm – những khái niệm, kiến thức và kỹ năng cơ bản ban đầu về kiểm thử phần mềm, về quy trình quản lý chất lượng, đảm bảo chất lượng phần mềm thông qua giáo trình (nội bộ) Kiểm thử và đảm bảo chất lượng phần mềm (Software Testing and Quality Assurance).

Giáo trình này với mục tiêu cung cấp cho sinh viên công nghệ phần mềm có được kiến thức và kỹ năng về việc kiểm thử phần mềm, các công đoạn kiểm thử, các loại kiểm thử, công cụ kiểm thử, xây dựng tài liệu kiểm thử, dữ liệu kiểm thử Và xây quy trình đảm bảo chất lượng phần mềm, giới thiệu tổng quan về hệ thống quản lý chất lượng, nguyên tắc, kỹ thuật ... để đảm bảo rằng dự án phần mềm sẽ chuyển giao cho khách hàng đúng hạn, đúng yêu cầu.

Đây là giáo trình sơ khởi, còn nhiều vấn đề chưa đi sâu phân tích và thực hiện, còn mang tính lý thuyết nhiều. Tác giả hy vọng bạn đọc đóng góp ý kiến để phiên bản 2 đáp ứng tốt hơn yêu cầu của nhiều độc giả, của sinh viên và kể cả những người đang công tác tại các phòng phát triển và đảm bảo chất lượng phần mềm.

CHƯƠNG 1: CÁC KHÁI NIỆM

1.1. Các định nghĩa

“Lỗi phần mềm là chuyện hiển nhiên của cuộc sống. Chúng ta dù cố gắng đến mức nào thì thực tế là ngay cả những lập trình viên xuất sắc nhất cũng không có thể lúc nào cũng viết được những đoạn mã không có lỗi. Tính trung bình, ngay cả một lập trình viên loại tốt thì cũng có từ 1 đến 3 lỗi trên 100 dòng lệnh. Người ta ước lượng rằng việc kiểm tra để tìm ra các lỗi này chiếm phân nửa khối lượng công việc phải làm để có được một phần mềm hoạt động được”. (*Software Testing Techniques, Second Edition, by Boris Beizer, Van Nostrand Reinhold, 1990, ISBN 1850328803*).

Trên đây là một nhận định về công việc kiểm nghiệm (testing) chương trình.

Thật vậy, ngày nay càng ngày các chương trình (các phần mềm) càng trở lên phức tạp và đồ sộ. Việc tạo ra một sản phẩm có thể bán được trên thị trường đòi hỏi sự nỗ lực của hàng chục, hàng trăm thậm chí hàng ngàn nhân viên. Số lượng dòng mã lên đến hàng triệu. Và để tạo ra một sản phẩm thì không phải chỉ do một tổ chức đứng ra làm từ đầu đến cuối, mà đòi hỏi sự liên kết, tích hợp của rất nhiều sản phẩm, thư viện lập trình, ... của nhiều tổ chức khác nhau... Từ đó đòi hỏi việc kiểm nghiệm phần mềm càng ngày càng trở nên rất quan trọng và rất phức tạp.

Song song với sự phát triển các công nghệ lập trình, các ngôn ngữ lập trình... thì các công nghệ và kỹ thuật kiểm nghiệm phần mềm ngày càng phát triển và mang tính khoa học. Bài tiểu luận này với mục đích là tập hợp, nghiên cứu, phân tích các kỹ thuật, các công nghệ kiểm nghiệm phần mềm đang được sử dụng và phát triển hiện nay.

1.1.1. Định nghĩa:

Việc kiểm nghiệm là quá trình thực thi một chương trình với mục đích là tìm ra lỗi. (Glen Myers)

Giải thích theo mục đích:

Việc thử nghiệm hiển nhiên là nói đến các lỗi (error), sai sót (fault), hỏng hóc (failure) hoặc các hậu quả (incident). Một phép thử là một cách chạy phần mềm theo các trường hợp thử nghiệm với mục tiêu là:

- Tìm ra sai sót.
- Giải thích sự hoạt động chính xác.

(Paul Jorgensen)

1.1.2. Các 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.
- **Hồng 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 hồng 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 hồng hóc và báo hiệu cho người dùng biết sự xuất hiện của hồng 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.
- **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.

So sánh giữa Thẩm tra và Xác nhận:

- **Thẩm tra:** thẩm tra quan tâm đến việc ngăn chặn lỗi giữa các công đoạn.
- **Xác nhận:** xác nhận quan tâm đến sản phẩm cuối cùng không còn lỗi.

1.2. Vòng đời của việc kiểm nghiệm (testing life cycle):

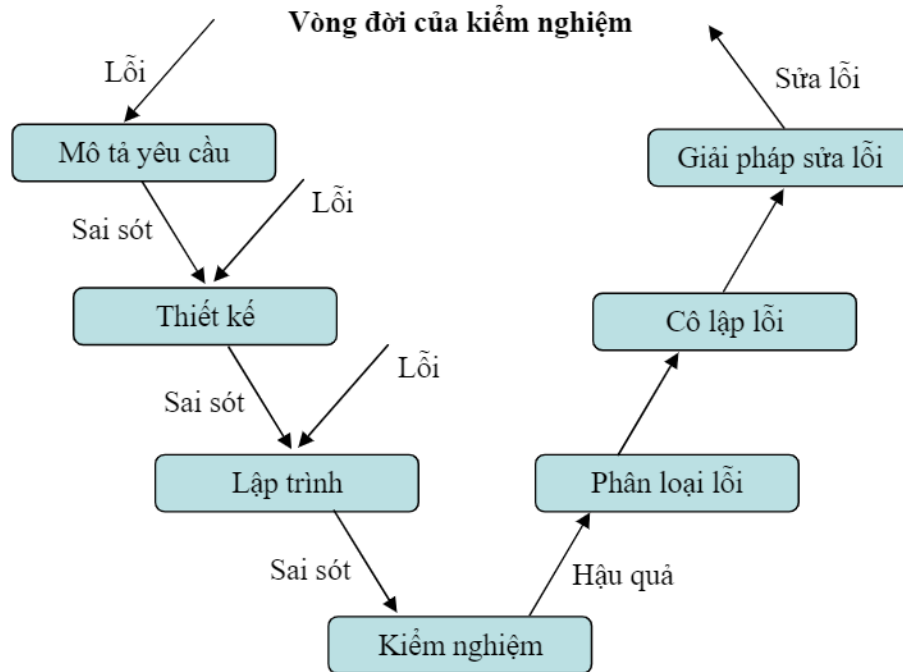
Bảng dưới đây mô tả các công đoạn phát triển một phần mềm và cách khắc phục lỗi.

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).

Đến công đoạn kiểm nghiệm 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.

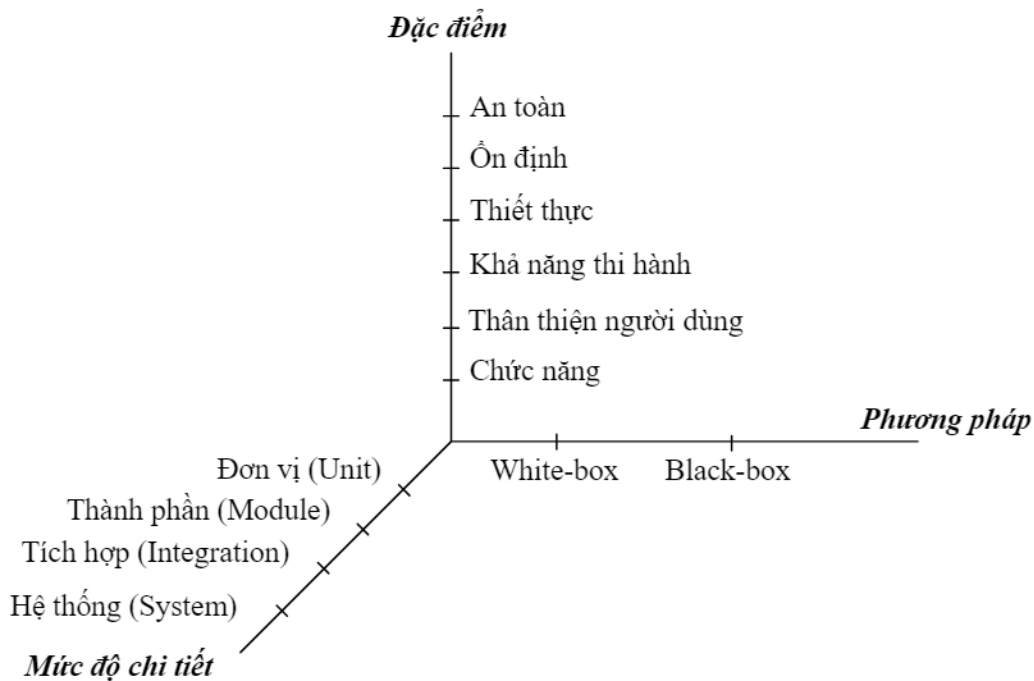


1.3. Phân loại kiểm nghiệm:

Có 2 mức phân loại:

- Một là 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)
- Cách phân loại khác là dựa trên phương pháp thử nghiệm (thường dùng ở mức kiểm tra đơn vị)
 - Kiểm nghiệm hộp đen (Black box testing) dùng để kiểm tra chức năng.
 - Kiểm nghiệm 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”

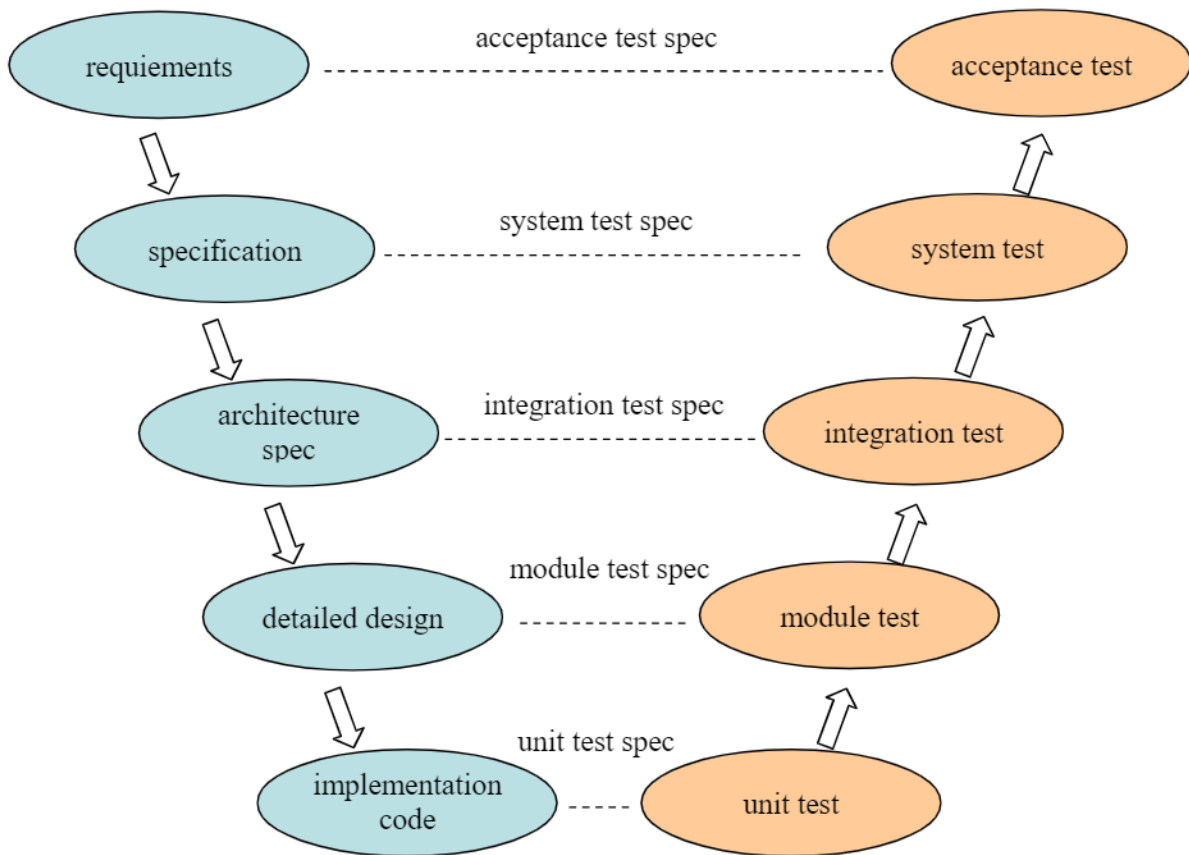


1.4. Sự tương quan giữa các công đoạn xây dựng phần mềm và loại kiểm nghiệm: Mô hình chữ V

Mô hình này nhằm giải thích sự tương quan giữa các công đoạn xây dựng phần mềm và các loại kiểm nghiệm. Ở mỗi công đoạn xây dựng phần mềm sẽ tương ứng với một loại kiểm nghiệm và cần có một hồ sơ kiểm nghiệm tương ứng được thành lập để phục vụ cho việc kiểm nghiệm.

Ví dụ:

- **Công đoạn:** Yêu cầu phần mềm (*requirements*); **Loại kiểm nghiệm:** Kiểm nghiệm chấp nhận (*acceptance test*); **Hồ sơ:** hồ sơ kiểm nghiệm chấp nhận (*acceptance test spec*).
- **Công đoạn:** Mô tả chi tiết phần mềm (*specification*); **Loại kiểm nghiệm:** Kiểm nghiệm hệ thống (*system test*); **Hồ sơ:** hồ sơ kiểm nghiệm hệ thống (*system test spec*).
- **Công đoạn:** Hồ sơ kiến trúc (*architecture spec*); **Loại kiểm nghiệm:** Kiểm nghiệm tích hợp (*integration test*); **Hồ sơ:** hồ sơ kiểm nghiệm tích hợp (*integration test spec*).
- **Công đoạn:** Thiết kế chi tiết (*detailed design*); **Loại kiểm nghiệm:** Kiểm nghiệm khối (*module test*); **Hồ sơ:** hồ sơ kiểm nghiệm khối (*module test spec*).
- **Công đoạn:** Viết mã (*implementation code*); **Loại kiểm nghiệm:** Kiểm nghiệm đơn vị (*unit test*); **Hồ sơ:** hồ sơ kiểm nghiệm đơn vị (*unit test spec*).



1.5. Sơ lược các kỹ thuật và công đoạn kiểm nghiệm:

Các kỹ thuật và công đoạn kiểm nghiệm có thể chia như sau:

- Kiểm nghiệm tầm hẹp: kiểm nghiệm các bộ phận riêng rẽ.
 - Kiểm nghiệm hộp trắng (White box testing)
 - Kiểm nghiệm hộp đen (Black box testing)
- Kiểm nghiệm tầm rộng:
 - Kiểm nghiệm bộ phận (Module testing): kiểm nghiệm một bộ phận riêng rẽ.
 - Kiểm nghiệm tích hợp (Integration testing): tích hợp các bộ phận và hệ thống con.
 - Kiểm nghiệm hệ thống (System testing): kiểm nghiệm toàn bộ hệ thống.
 - Kiểm nghiệm chấp nhận (Acceptance testing): thực hiện bởi khách hàng.

1.5.1. Các loại kiểm nghiệm tầm hẹp:

Các loại kiểm nghiệm này được thực hiện để kiểm nghiệm đến các đơn vị (unit) hoặc các khối chức năng (module).

a. Kiểm nghiệm hộp trắng (white-box testing)

Còn gọi là kiểm nghiệm cấu trúc. Kiểm nghiệm theo cách này là loại kiểm nghiệm sử dụng các thông tin về cấu trúc bên trong của ứng dụng. Việc kiểm nghiệm này dựa trên quá trình thực hiện xây dựng phần mềm.

Tiêu chuẩn của kiểm nghiệm 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 nghiệm hộp đen (black-box testing)

Còn gọi là kiểm nghiệm chức năng. Việc kiểm nghiệm này được thực hiện mà không cần quan tâm đến các thiết kế và viết mã của chương trình. Kiểm nghiệm theo cách này chỉ quan tâm đến chức năng đã đề ra của chương trình. Vì vậy kiểm nghiệm 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 nghiệm 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 nghiệm tại biên:

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

Ví dụ:

if $x > y$ then S1 else S2

Với điều kiện bao phủ, chỉ cần 2 trường hợp thử là $x > y$ và $x \leq y$.

Với kiểm nghiệm đường biên thì kiểm tra với các trường hợp thử là $x > y$, $x < y$, $x = y$

Các loại kiểm nghiệm tầm rộng:

Việc kiểm nghiệm này thực hiện trên tầm mức lớn hơn và các khía cạnh khác của phần mềm như kiểm nghiệm hệ thống, kiểm nghiệm sự chấp nhận (của người dùng)...