

## Contents

<b>C</b>	1
correctness 166 (tính đúng đắn) :	1
correctness proof 167 (bằng chứng tính đúng đắn):	1
<b>D</b>	1
defect 155 (khuyết điểm):	1
desk check 175 (bàn kiểm tra):	1
Luật DIJkstra là gì ?:	1
Tại sao luật DIJkstra lại đúng?	2
<b>E</b>	2
error 155(lỗi):	2
execution-based testing 163 (kiểm tra dựa trên thực thi)	2
<b>F</b>	2
failure 155 (lỗi)	2
fault 155 (lỗi)	3
fault density 162 (mật độ lỗi)	3
fault detection efficiency 162 (hiệu quả phát hiện lỗi)	3
fault detection rate 162 (tỷ lệ phát hiện lỗi)	3
follow-up 160 (quá trình theo dõi)	3
<b>I</b>	3
inspection 159 ( điều tra)	3
inspection rate 162 ( tỉ lệ kiểm tra)	3
<b>L</b>	4
loop invariant 169 ( vòng lặp bất biến)	4
<b>M</b>	4
managerial independence 156 ( Độc lập quản lý)	4
mean time between failures (MTBF) 164 ( thời gian trung bình giữa các lần thất bại)	4
mean time to repair (MTTF) 164 (thời gian sửa chữa)	4
mistake 155 (Lỗi)	5
model checking 174 (Kiểm tra mô hình)	5
moderator 160 (Người điều hành)	5
<b>N</b>	5
non-execution-based testing 157 (Kiểm thử không dựa trên thực thi)	5
<b>O</b>	5
overview 159 (Tổng quan)	5

<b>P</b>	5
performance 165 (Hiệu suất)	5
preparation 159 (Sự chuẩn bị)	6
<b>Q</b>	6
quality 156 (Chất lượng phần mềm) :	6
Quality assurance 157 (đảm bảo chất lượng) :	6
Quality assurance thì khác gì với testing?	6
<b>R</b>	7
reader 160 :	7
recorder 160:	7
regression testing (Kiểm tra hồi quy) 176	7
reliability (Độ tin cậy) 164:	7
rework (Làm lại) 160	7
robustness (Tính Ổn định) 165	7
<b>S</b>	8
simulator (Trình mô phỏng) 164	8
SQA - software quality assurance (Đảm bảo chất lượng phần mềm) 156	8
Nếu nhóm SQA phát hiện ra ít lỗi, thì có thể kết luận nhóm code giỏi hay nhóm SQA kém? Tại sao? *	9
Người ta nói “Nhóm SQA tạo ra chất lượng cho phần mềm”. Đúng hay sai? Tại sao? *	9
systematic testing (Kiểm tra hệ thống) 175	9
<b>T</b>	9
test workflow 155 Quy trình kiểm tra	9
testing 155 (Kiểm thử)	10
<b>U</b>	10
utility 164 (Tính hữu dụng)	10
<b>V</b>	10
Thẩm định và xác minh là gì (V&V)? Tầm quan trọng của chúng?	10
Có những loại V&V nào? Mô tả nội dung mỗi loại	11
validation 155	11
verification 155	12
<b>W</b>	12
walkthrough 158:	12
Tại sao trong nhóm walkthrough và inspection, luôn phải có đại diện của workflow tiếp theo?:	12
Tại sao nói inspection và walkthrough là hướng tài liệu, mà không phải hướng vào người tham gia?	13

Tại sao người phụ trách nhóm inspection hay walkthrough phải là người của nhóm SQA?	13
So sánh walkthrough và inspection ?	13
Phần mềm còn lỗi hay không khi thực hiện hết các ca kiểm thử được thiết kế? Khi nào dừng kiểm thử?	14
Kiểm thử hệ thống nhằm kiểm tra cái gì? Ai thực hiện? Các phương pháp?	15
Trình bày các kiểm thử được thực hiện trong kiểm thử hệ thống?	15
Kiểm thử chấp nhận là gì? Trong đó có những kiểm thử nào được thực hiện? Phân biệt.	16
Test trường hợp sai kiểu dữ liệu đầu vào thì thuộc thể loại test gì?	17

## C

### **correctness 166 (tính đúng đắn) :**

Nghĩa là sản phẩm được triển khai chính xác đặc điểm kỹ thuật của nó. Nói cách khác, có nhiều thứ để kiểm tra hơn là chỉ cho thấy rằng sản phẩm là chính xác.

### **correctness proof 167 (bằng chứng tính đúng đắn):**

là một kỹ thuật toán học để chỉ ra rằng một sản phẩm là chính xác, nói cách khác, nó đáp ứng các thông số kỹ thuật của nó. Kỹ thuật này đôi khi được gọi là xác minh. Tuy nhiên, như đã chỉ ra trước đây, thuật ngữ này có một ý nghĩa khác trong ngữ cảnh thử nghiệm. Ngoài ra, xác minh cũng thường được sử dụng để biểu thị tất cả các kỹ thuật, không chỉ chứng minh tính đúng đắn. Để rõ ràng, quy trình toán học này sẽ là gọi là chứng minh tính đúng đắn, để nhắc nhở người đọc rằng đó là một quá trình chứng minh toán học.

## D

### **defect 155 (khuyết điểm):**

Lỗi trong quá trình phát triển hoặc lỗi logic(coding or logic) làm cho chương trình hoạt động sai yêu cầu đề ra. (Về cơ bản là giống định nghĩa bug).

### **desk check 175 (bàn kiểm tra):**

là quá trình tự rà soát mã nguồn của một chương trình. Nó liên quan đến việc đọc qua các chức năng bên trong mã và tự thử nghiệm chúng, thường có nhiều giá trị đầu vào. Các nhà phát triển có thể bàn kiểm tra mã của họ trước khi phát hành một chương trình phần mềm để đảm bảo các thuật toán được hoạt động một cách hiệu quả và chính xác.

**Luật Dijkstra là gì ? : là rất dễ để chứng minh 1 phần mềm có lỗi, nhưng không thể chứng minh được 1 phần mềm ko có lỗi**

Theo Luật Dijkstra :

- Thử nghiệm chương trình có thể là một cách rất hiệu quả để chỉ ra sự hiện diện của lỗi, nhưng nó không đủ khả năng để cho thấy sự vắng mặt của chúng, nếu một sản phẩm được thực thi với dữ liệu thử nghiệm và kết quả đầu ra bị sai, thì sản phẩm chắc chắn có lỗi. Nhưng nếu đầu ra là chính xác, sau đó vẫn có thể có lỗi trong sản phẩm; thông tin duy nhất mà có thể được suy ra từ thử nghiệm cụ thể đó là sản phẩm chạy chính xác trên tập hợp dữ liệu thử nghiệm
- Lập trình viên nên để chương trình chứng minh và chương trình phát triển cùng nhau.
- Cách hiệu quả duy nhất để nâng cao mức độ tin cậy của một chương trình có ý nghĩa quan trọng là đưa ra một bằng chứng thuyết phục về tính đúng đắn của nó. Nhưng ngay cả khi một sản phẩm được chứng minh là đúng, nó phải được kiểm tra kỹ lưỡng như ỏn

**Xem nội dung đầy đủ tại:**

<https://123docz.net//document/2961752-giai-ngan-hang-cong-nghe-phan-mem-ptit.htm>

**Tại sao luật Dijkstra lại đúng?**

nếu một sản phẩm được thực thi với dữ liệu thử nghiệm và kết quả đầu ra bị sai, thì sản phẩm chắc chắn có lỗi. Nhưng nếu đầu ra là chính xác, sau đó vẫn có thể có lỗi trong sản phẩm; thông tin duy nhất mà có thể được suy ra từ thử nghiệm cụ thể đó là sản phẩm chạy chính xác trên tập hợp dữ liệu thử nghiệm.

Cách hiệu quả duy nhất để nâng cao mức độ tin cậy của một chương trình có ý nghĩa quan trọng là đưa ra một bằng chứng thuyết phục về tính đúng đắn của nó. Nhưng ngay cả khi một sản phẩm được chứng minh là đúng, nó phải được kiểm tra kỹ lưỡng

*Hầu hết các sản phẩm ngày nay rất đa dạng và phức tạp do được phát triển trên nhiều nền tảng, công nghệ phong phú cũng như khả năng lưu trữ kết nối dữ liệu lớn, khiến việc kiểm thử trở nên khó khăn và việc kiểm thử toàn bộ là gần như không thể. Kiểm thử với tất cả các kết hợp đầu vào và đầu ra, với tất cả các kịch bản là không thể trừ phi nó chỉ bao gồm ít trường hợp thì có thể kiểm thử toàn bộ.*

## E

### **error 155(lỗi):**

Là hành động của con người dẫn đến kết quả sai.

### **execution-based testing 163 (kiểm tra dựa trên thực thi)**

là một quá trình suy ra các thuộc tính hành vi nhất định của một sản phẩm, một phần dựa trên kết quả của việc thực thi sản phẩm trong một môi trường cho trước với các input đã được chọn lọc.

## F

### **failure 155 (lỗi)**

là hành vi không chính xác có thể quan sát thấy của sản phẩm phần mềm như là kết quả của 1 lỗi (fault) và lỗi là số lượng kết quả không chính xác. Một lỗi (failure) cụ thể có thể do một số lỗi (fault) gây ra và một số lỗi (fault) có thể không bao giờ gây ra lỗi (failure).

### **fault 155 (lỗi)**

là một lỗi được đưa vào phần mềm khi con người gây ra 1 sai sót nào đó. Một sai sót của phần mềm chuyên dụng có thể gây ra nhiều lỗi; ngược lại, các sai sót khác nhau có thể gây ra lỗi giống hệt nhau.

### **fault density 162 (mật độ lỗi)**

được đo bằng số lỗi trên mỗi trang được kiểm tra hoặc số lỗi trên 1000 dòng mã (KLOC) được kiểm tra. Số liệu này có thể được chia nhỏ thành các lỗi lớn trên một đơn vị vật liệu và các lỗi nhỏ trên một đơn vị vật liệu.

### **fault detection efficiency 162 (hiệu quả phát hiện lỗi)**

tức là số lỗi lớn và nhỏ được phát hiện trên mỗi người-giờ.

### **fault detection rate 162 (tỷ lệ phát hiện lỗi)**

tức là số lỗi lớn và nhỏ được phát hiện mỗi giờ.

### **follow-up 160 (quá trình theo dõi)**

là quá trình mà người kiểm duyệt phải đảm bảo rằng mọi vấn đề nêu ra đã được giải quyết ổn thỏa, bằng cách sửa tài liệu hoặc làm rõ các mục bị gấn cờ không chính xác là lỗi. Tất cả các bản sửa lỗi phải được kiểm tra để đảm bảo rằng không có lỗi mới nào được đưa ra.

## I

### inspection 159 ( điều tra)

là kiểm tra định kì các hoạt động cụ thể để thử nghiệm các thiết kế và mã. Một cuộc thanh tra vượt xa một hướng dẫn và có năm bước chính thức. Việc kiểm tra nên được tiến hành bởi một nhóm bốn người.

Ví dụ, trong trường hợp kiểm tra thiết kế, nhóm bao gồm người kiểm duyệt, người thiết kế, người thực hiện và người kiểm tra. Người điều hành vừa là người quản lý vừa là trưởng đoàn thanh tra. Phải có đại diện của nhóm chịu trách nhiệm về quy trình công việc hiện tại cũng như đại diện của nhóm chịu trách nhiệm về quy trình công việc tiếp theo.

### inspection rate 162 ( tỉ lệ kiểm tra)

Khi các thông số kỹ thuật và thiết kế được kiểm tra, số lượng trang được kiểm tra mỗi giờ có thể được đo lường; để kiểm tra mã, một số liệu thích hợp là các dòng mã được kiểm tra mỗi giờ

## L

### loop invariant 169 ( vòng lặp bất biến)

là một số vị ngữ (điều kiện) giữ cho mỗi lần lặp của vòng lặp.

+Chú ý: Có một điều mà nhiều người không nhận ra ngay lập tức khi xử lý các vòng lặp và bất biến. Chúng bị lẫn lộn giữa bất biến vòng lặp và vòng lặp có điều kiện (điều kiện kiểm soát sự kết thúc của vòng lặp). Như mọi người chỉ ra, “ *bất biến vòng “lặp* phải đúng

1. trước khi vòng lặp bắt đầu
2. trước mỗi lần lặp của vòng lặp
3. sau khi vòng lặp kết thúc

(mặc dù nó có thể tạm thời là sai trong phần thân của vòng lặp). *Mặt khác, “ điều kiện vòng lặp “* phải là sai sau khi vòng lặp kết thúc, nếu không thì vòng lặp sẽ không bao giờ chấm dứt.

-> Do đó, bất biến vòng lặp và điều kiện vòng lặp phải là các điều kiện khác nhau.

## M

### managerial independence 156 ( Độc lập quản lý)

: là sự độc lập về mặt quản lý giữa nhóm phát triển và nhóm SQA. Có nghĩa là, sự phát triển nên được đặt dưới một người quản lý, SQA dưới một người quản lý khác và không người quản lý nào có thể thay thế người kia.

### **mean time between failures (MTBF) 164 ( thời gian trung bình giữa các lần thất bại)**

là thời gian trung bình từ thất bại này đến thất bại tiếp theo. Thông thường mọi người sẽ nghĩ về nó như thời gian trung bình mà một cái gì đó hoạt động cho đến khi nó thất bại và cần phải được sửa chữa (một lần nữa).

- MTBF dùng để làm gì.? **MTBF là thước đo độ tin cậy của thiết bị**, có thể được sử dụng để xác định lịch bảo trì, để xác định có bao nhiêu phụ tùng cần có sẵn để sửa chữa lỗi trong một nhóm thiết bị. Để tính toán MTBF, chúng ta cần biết tổng số giờ đơn vị thử nghiệm được thực hiện trong quá trình thử nghiệm và số lần thất bại xảy ra.

### **mean time to repair (MTTF) 164 (thời gian sửa chữa)**

là thời gian sửa chữa trung bình là bao lâu. Nhưng, thường quan trọng hơn là mất bao lâu để sửa chữa kết quả của lỗi.

### **mistake 155 (Lỗi)**

Nói về một hành động hoặc một ý nghĩa sai lầm, đem lại một kết quả không như mong muốn ( chủ yếu được thực hiện bởi một người)

### **model checking 174 (Kiểm tra mô hình)**

Kiểm tra mô hình là một công nghệ mới có thể thay thế cho việc chứng minh tính đúng đắn của phần mềm.

### **moderator 160 (Người điều hành)**

Khi bắt đầu kiểm tra, một người tham gia cùng đoàn kiểm tra xem qua tài liệu, đảm bảo rằng mọi hạng mục đều được che đậy và mọi nhánh đều được thực hiện ít nhất một lần. Sau đó, việc tìm kiếm lỗi bắt đầu. Như với hướng dẫn, mục đích là để tìm và ghi lại các lỗi, không phải để sửa chúng. Trong thời hạn một ngày, Trưởng đoàn kiểm tra (người điều hành) phải báo cáo bằng văn bản về việc kiểm tra để đảm bảo việc theo dõi một cách tỉ mỉ.

## **N**

### **non-execution-based testing 157 (Kiểm thử không dựa trên thực thi)**

Việc kiểm thử phần mềm mà không chạy các trường hợp kiểm thử được gọi là kiểm thử không dựa trên thực thi. Ví dụ về các phương pháp kiểm tra không dựa trên thực thi bao gồm xem xét phần mềm (đọc kĩ phần mềm) và phân tích phần mềm theo phương pháp toán học

## O

### **overview 159 (Tổng quan)**

Tổng quan về tài liệu được kiểm tra (yêu cầu, đặc điểm cụ thể, thiết kế, mã hoặc kế hoạch) được cung cấp bởi một trong những cá nhân chịu trách nhiệm sản xuất tài liệu đó. Vào cuối phần tổng quan, tài liệu được phát cho các học viên.

## P

### **performance 165 (Hiệu suất)**

Hiệu suất là một khía cạnh khác của sản phẩm phải được kiểm tra. Ví dụ, điều cần thiết là phải biết mức độ mà sản phẩm đáp ứng các hạn chế của nó liên quan đến các yêu cầu về thời gian hoặc không gian đáp ứng. Đối với một hệ thống máy tính như chẳng hạn như một máy tính tích hợp trong tên lửa phòng không cầm tay, các hạn chế về không gian của hệ thống có thể là chỉ 128 megabyte (MB) bộ nhớ chính có sẵn cho phần mềm. Cho dù phần mềm có xuất sắc đến đâu, nhưng nếu nó cần 256 MB bộ nhớ chính, thì nó không thể sử dụng được.

### **preparation 159 (Sự chuẩn bị)**

Trong quá trình chuẩn bị, những người tham gia cố gắng đọc hiểu tài liệu một cách chi tiết. Danh sách các loại lỗi được tìm thấy trong các cuộc kiểm tra gần đây, với các loại lỗi được xếp hạng theo tần suất, là những trợ giúp tuyệt vời. Những danh sách này giúp các thành viên trong nhóm tập trung vào các vùng mà lỗi xảy ra nhiều nhất

## Q

### **quality 156 (Chất lượng phần mềm) :**

là mức độ mà phần mềm đáp ứng các thông số, yêu cầu kỹ thuật của chính nó. Tuy nhiên, để kết luận sản phẩm có đạt yêu cầu hay không chỉ chất lượng thôi là chưa đủ.

Ví dụ, để đảm bảo rằng một sản phẩm có thể được bảo trì dễ dàng, sản phẩm phải được thiết kế tốt và lập trình 1 cách tỉ mỉ. Do vậy, rất cần thiết để tạo ra 1 sản phẩm có chất lượng cao, nhưng để kết luận sản phẩm đó là đạt yêu cầu chưa là chưa đủ.

### **Quality assurance 157 (đảm bảo chất lượng) :**

- là việc ngăn ngừa những khiếm khuyết của sản phẩm để đảm bảo chất lượng dựa trên những phương pháp, kỹ thuật, quy trình và cách tiếp cận được thiết kế một cách chính xác cho từng dự án.
- quality assurance chủ động phòng chống những khiếm khuyết và sai sót để đảm bảo chất lượng đầu ra của sản phẩm phù hợp với nhu cầu, nguyện vọng của khách hàng



### **Quality assurance thì khác gì với testing?**

- Quality assurance là 1 chủ đề rộng hơn testing. Nó không chỉ quan tâm đến đầu ra của sản phẩm mà nó còn quan tâm cả đầu vào (cách sản phẩm được phát triển) để cải thiện tiến trình sản phẩm.
- Quality assurance thì tập trung vào quy trình và thủ tục hơn là tiến hành các thử nghiệm thực tế trên hệ thống. Còn testing thì ngược lại, tập trung vào các thử nghiệm thực tế
- Testing chỉ là việc đảm bảo phần mềm hoạt động chính xác và không gặp lỗi, còn quality assurance ngoài việc đảm bảo phần mềm hoạt động chính xác và không gặp lỗi ra thì quá trình này cần phải đảm bảo sao cho phần mềm đáp ứng thông số kỹ thuật tốt nhất có thể và làm cho khách hàng cảm thấy hài lòng.
- Mục đích Testing là kiểm soát chất lượng còn mục đích của quality assurance là đảm bảo chất lượng.

## **R**

### **reader 160 :**

là 1 chức vụ có vai trò đặc biệt của nhóm trong quá trình inspections. Nó do người có vai trò điều hành đảm nhiệm. Công việc của 1 reader là hướng dẫn, dẫn dắt nhóm trong khâu kiểm tra thiết kế.

### **recorder 160:**

là 1 chức vụ có vai trò đặc biệt của nhóm quá trình inspections. Nó do người có vai trò điều hành đảm nhiệm. Công việc của 1 recorder là tạo ra các báo cáo về những lỗi đã phát hiện được trong quá trình kiểm tra.

### **regression testing (Kiểm tra hồi quy) 176**

- Là quy trình kiểm tra sản phẩm được thử nghiệm dựa trên các trường hợp thử nghiệm trước đó để đảm bảo rằng chức năng của phần còn lại của sản phẩm không bị ảnh hưởng
- Khi một khi lập trình viên đã xác định rằng các thay đổi mong muốn đã được thực hiện, sản phẩm phải được thử nghiệm dựa trên các trường hợp thử nghiệm trước đó để đảm bảo rằng chức năng của phần còn lại của sản phẩm không bị ảnh hưởng. Thủ tục này được gọi là kiểm tra hồi quy

### **reliability (Độ tin cậy) 164:**

- Độ tin cậy là một thuộc tính của bất kỳ thành phần nào liên quan đến máy tính (ví dụ: phần mềm hoặc phần cứng hoặc mạng) hoạt động nhất quán theo các thông số kỹ thuật của nó.

- Nó là 1 thước đo tần suất và mức độ nghiêm trọng của lỗi sản phẩm- 1 điều không thể chấp nhận được đối với 1 sản phẩm đang trong quá trình vận hành.
- Do đó xác định độ tin cậy để cho ta biết tần suất gặp lỗi của sản phẩm và những ảnh hưởng mà nó gây ra.

### **rework (Làm lại) 160**

Là bước thứ 4 trong 5 bước Inspections. Trong quá trình làm lại, mỗi cá nhân chịu trách nhiệm về tài liệu sẽ giải quyết tất cả các lỗi trong tài liệu mình rà soát.

### **robustness (Tính Ổn định) 165**

Là một trong các khía cạnh yêu cầu trong kiểm thử. Kiểm tra khả năng chống lỗi/sự cố của sản phẩm.

Mặc dù rất khó để đưa ra một định nghĩa chính xác, nhưng về cơ bản, tính Ổn định là hàm của một số yếu tố, chẳng hạn như phạm vi điều kiện hoạt động, khả năng xảy ra kết quả không được chấp nhận với đầu vào hợp lệ và khả năng chấp nhận các tác động khi sản phẩm đầu vào không hợp lệ.

## **S**

### **simulator (Trình mô phỏng) 164**

Trình mô phỏng giúp một hệ thống thời gian thực có thể được kiểm tra. Trình mô phỏng là một mô hình hoạt động của môi trường mà phần mềm điều khiển thực thi.

Một trình mô phỏng có thể là một mô hình gần đúng của một mô hình thực về một số khía cạnh của hệ thống; nó không bao giờ có thể là chính hệ thống. Sử dụng trình mô phỏng có nghĩa là, trong khi thực sự có một “môi trường đã biết”, thì có rất ít khả năng rằng môi trường đã biết này giống với môi trường thực tế mà sản phẩm sẽ được cài đặt theo.

*Bonus: ví dụ - phần mềm điều khiển thực thi bay: được kiểm tra bằng cách làm cho trình mô phỏng gửi các đầu vào đã chọn đến phần mềm điều khiển chuyển bay. Trình mô phỏng có các điều khiển cho phép người vận hành đặt biến đầu vào thành bất kỳ giá trị đã chọn nào.*

*Nếu mục đích của thử nghiệm là xác định cách phần mềm điều khiển chuyển bay hoạt động nếu một động cơ bốc cháy, thì các điều khiển của bộ mô phỏng được thiết lập để đầu vào được gửi đến phần mềm điều khiển chuyển bay không thể phân biệt được với đầu vào sẽ được gửi nếu một động cơ của máy bay thực tế đã bốc cháy. Đầu ra được phân tích bằng cách kiểm tra các tín hiệu đầu ra được gửi từ phần mềm điều khiển chuyển bay đến bộ mô phỏng.*

## **SQA - software quality assurance (Đảm bảo chất lượng phần mềm) 156**

- Vai trò của nhóm SQA:

- + đảm bảo rằng các nhà phát triển đang thực sự làm công việc chất lượng cao
- + kiểm tra xem sản phẩm của nhà phát triển có chính xác hay không
- + đảm bảo chất lượng của quy trình phần mềm và do đó đảm bảo chất lượng của sản phẩm.

- Đặc điểm:

- + Nhóm SQA luôn mong muốn đảm bảo sản phẩm khi được giao cho khách hàng là không có lỗi nên họ thường cầu toàn, không chấp nhận việc sản phẩm còn lỗi, nếu còn lỗi thì muốn phải bàn giao thêm thời gian để test thêm.
- + Nhóm SQA làm việc độc lập và ngang hàng với nhóm phát triển

**Nếu nhóm SQA phát hiện ra ít lỗi, thì có thể kết luận nhóm code giỏi hay nhóm SQA kém? Tại sao? \***

*(Câu này không tìm được, trả lời theo ý hiểu)*

Không thể đưa ra kết luận ngay trình độ của các nhóm vì hoàn toàn có thể là nhóm code giỏi nên có ít lỗi, hoặc cũng có thể do nhóm SQA không phát hiện được lỗi.

Tuy nhiên, nếu dựa vào phạm vi của dự án, nếu dự án có quy mô lớn, phức tạp thì việc phát sinh ra ít lỗi là ít gặp nên nhóm SQA cũng phải phát hiện được nhiều lỗi, nếu khi đó vẫn phát hiện ít lỗi thì phần trăm cao là do nhóm SQA kém hơn.

**Người ta nói “Nhóm SQA tạo ra chất lượng cho phần mềm”. Đúng hay sai? Tại sao? \***

Sai, SQA không tạo ra chất lượng phần mềm, họ chỉ bảo đảm chất lượng phần mềm. Họ đảm bảo rằng tất cả các quy trình, phương pháp, hoạt động và hạng mục công việc kỹ thuật phần mềm đều được giám sát và tuân thủ theo các tiêu chuẩn đã xác định. Còn người tạo nên chất lượng phần mềm là Developer.

*(Theo ngân hàng)*

## **systematic testing (Kiểm tra hệ thống) 175**

Khi người lập trình hài lòng rằng mã tạo tác hoạt động chính xác, quá trình kiểm tra hệ thống sẽ bắt đầu. Một phần thiết yếu của trường hợp thử nghiệm là một tuyên bố về kết quả đầu ra mong đợi trước khi thử nghiệm được thực hiện.

Người thực hiện không nên là người phát triển vì khi thực hiện kiểm thử, người thực hiện luôn trong tâm thế “sản phẩm đã chính xác” do họ chính là người thực hiện, đầu vào và đầu ra thường trả về kết quả mà họ mong muốn.

Vì vậy người kiểm tra phải là một nhóm khác - nhóm SQA, họ có cái nhìn tổng quát các trường hợp xảy ra và mục tiêu là đảm bảo chất lượng phần mềm.

**T**

### **test workflow 155 Quy trình kiểm tra**

là quy trình các chuyên gia phần mềm phải kiểm tra và thử nghiệm lại từng hiện vật mà họ phát triển hoặc duy trì. Sau đó được chuyển giao cho nhóm đảm bảo chất lượng phần mềm để kiểm tra độc lập. Đây cũ quy trình diễn ra song song với các quy trình khác từ lúc bắt đầu tới lúc kết thúc, có 2 khía cạnh chính của kiểm thử:

- Mọi nhà phát triển và bảo trì phải đảm bảo rằng công việc của họ làm là chính xác. Do đó, các chuyên gia về phần mềm phải kiểm tra hoặc kiểm tra mọi artifact mà họ phát triển hoặc bảo trì
- Một khi họ chắc chắn rằng artifact của họ đã hoạt động tốt, nó sẽ được chuyển tới nhóm đảm bảo chất lượng phần mềm để kiểm thử độc lập.

Mục đích chính của quy trình kiểm thử là đảm bảo tất cả các artifact hoạt động tốt và không gây ra ảnh hưởng nào nghiêm trọng tới các artifact liên quan đến nó.

### **testing 155 (Kiểm thử)**

Kiểm thử (testing) là một quá trình đánh giá một hệ thống hay là các thành phần của nó với mục đích là xác định xem nó có thỏa mãn những yêu cầu được đưa ra hay không, có sự khác biệt nào giữa phần mềm thực tế đang tồn tại và những điều kiện được yêu cầu (requirement).

Hiểu một cách đơn giản, kiểm thử – test là chạy một quá trình để xác nhận bất kỳ thiếu sót (defect), lỗi (bug), sai sót (error) hay những yêu cầu bị bỏ quên, những yêu cầu không đúng so với yêu cầu thực tế đề ra

**U**

### **utility 164 (Tính hữu dụng)**

Tiện ích là mức độ mà nhu cầu của người dùng được đáp ứng khi một sản phẩm phù hợp được sử dụng theo điều kiện cho phép của các thông số kỹ thuật của nó. Nói cách khác, một sản phẩm đang hoạt động chính xác bây giờ là tùy thuộc vào các đầu vào hợp lệ về các thông số kỹ thuật

Tiện ích của sản phẩm nên được thử nghiệm trước, và nếu sản phẩm không đạt điểm đó, việc thử nghiệm sẽ dừng lại.

## V

### **Thẩm định và xác minh là gì (V&V)? Tầm quan trọng của chúng?**

Verification & Validation hay còn là Thẩm định và Kiểm định, trong đó

- Thẩm định là xác định xem hệ thống có phù hợp với yêu cầu và thực hiện các chức năng mà nó được dự định và đáp ứng các mục tiêu của tổ chức và nhu cầu của người dùng hay không. Thẩm định được thực hiện vào cuối của quá trình phát triển và diễn ra sau khi việc kiểm định được hoàn thành và sau khi một sản phẩm được xây dựng theo các tiêu chí nhằm đảm bảo rằng các sản phẩm tích hợp một cách chính xác vào môi trường.
- Kiểm định là để chắc chắn rằng sản phẩm được thiết kế để cung cấp tất cả các chức năng cho khách hàng. Kiểm định được thực hiện từ lúc bắt đầu của quá trình phát triển phần mềm. Nó bao gồm các đánh giá và các cuộc họp, rà soát, kiểm tra, ... để đánh giá tài liệu, kế hoạch, việc lập trình, các yêu cầu và các thông số kỹ thuật. Được thực hiện trong suốt quá trình phát triển dựa trên những thao tác chủ chốt như rà soát, đánh giá và kiểm tra, phản hồi của người cố vấn, đào tạo, bản danh sách và tiêu chuẩn..
- Mục đích của việc này là xác định xem sản phẩm đã thỏa mãn hoàn toàn bản đặc tả chưa

### **Có những loại V&V nào? Mô tả nội dung mỗi loại**

- Vì V&V thực hiện thẩm định vào cuối mỗi Workflow, trong khi yêu cầu kiểm tra được thực hiện thường xuyên trong quá trình phát triển sản phẩm và hoạt động bảo trì là cần thiết. Cho nên người ta thường sử dụng “testing” thay cho V&V.
- Có 2 loại testing là : execution-based testing(test thực thi) and non-execution based testing(test không cần thực thi).
- execution-based testing: Sử dụng đầu vào định sẵn, với nền tảng, môi trường định sẵn. Thực hiện suy diễn từ đầu vào ra kết quả thực thi. (vd: viết test - case chạy code)
- non-execution based testing: kiểm tra phần mềm không sử dụng test-case. (vd: đọc lại source code phần mềm, phân tích logic toán học trong chương trình ). Gồm có 2 loại là WalkThrough và Inspection.

### **validation 155**

Thẩm định là xác định xem hệ thống có phù hợp với yêu cầu và thực hiện các chức năng mà nó được dự định và đáp ứng các mục tiêu của tổ chức và nhu cầu của người dùng hay

không. Thử nghiệm được thực hiện vào cuối của quá trình phát triển và diễn ra sau khi việc kiểm định được hoàn thành và sau khi một sản phẩm được xây dựng theo các tiêu chí nhằm đảm bảo rằng các sản phẩm tích hợp một cách chính xác vào môi trường. Đây là một hoạt động cấp cao. Nó xác định tính đúng đắn của sản phẩm cuối cùng của một dự án phát triển đối với các nhu cầu và yêu cầu của người sử dụng.

Ưu điểm của kiểm định phần mềm:

- Kiểm định giúp hạ thấp các khiếm khuyết trong các giai đoạn phát triển sau này.
- Kiểm định các sản phẩm ở giai đoạn khởi đầu của sự phát triển sẽ giúp hiểu rõ sản phẩm một cách tốt hơn.
- Nó làm giảm nguy cơ thất bại trong các ứng dụng phần mềm hoặc ứng dụng sản phẩm.
- Nó giúp việc xây dựng các sản phẩm đúng theo các thông số kỹ thuật của khách hàng và nhu cầu.

#### **verification 155**

Kiểm định là để chắc chắn rằng sản phẩm được thiết kế để cung cấp tất cả các chức năng cho khách hàng. Kiểm định được thực hiện từ lúc bắt đầu của quá trình phát triển phần mềm. Nó bao gồm các đánh giá và các cuộc họp, rà soát, kiểm tra, ... để đánh giá tài liệu, kế hoạch, việc lập trình, các yêu cầu và các thông số kỹ thuật. Được thực hiện trong suốt quá trình phát triển dựa trên những thao tác chủ chốt như rà soát, đánh giá và kiểm tra, phản hồi của người cố vấn, đào tạo, bản danh sách và tiêu chuẩn. Đây là hoạt động cấp thấp. Nó thể hiện tính thống nhất, đầy đủ và chính xác của phần mềm ở từng giai đoạn và giữa mỗi giai đoạn của vòng đời phát triển.

Ưu điểm của kiểm định phần mềm:

- Kiểm định giúp hạ thấp các khiếm khuyết trong các giai đoạn phát triển sau này.
- Kiểm định các sản phẩm ở giai đoạn khởi đầu của sự phát triển sẽ giúp hiểu rõ sản phẩm một cách tốt hơn.
- Nó làm giảm nguy cơ thất bại trong các ứng dụng phần mềm hoặc ứng dụng sản phẩm.
- Nó giúp việc xây dựng các sản phẩm đúng theo các thông số kỹ thuật của khách hàng và nhu cầu.

#### **W**

#### **walkthrough 158:**

Gồm có 4-6 người, 1 người viết đặc tả (specification), 1 người quản lý workflow phân tích, đại diện khách hàng, 1 người thuộc team thiết kế, 1 đại diện thuộc SQA (leader).

là một Kỹ thuật đánh giá không chính thức (nên không có quản lý, giám đốc dự án). Những người tham gia phải xem tài liệu trước cuộc họp (ít nhất vài ngày). Tác giả giải thích tài liệu/ sản phẩm đó cho nhóm (tác giả, điều phối viên, giám định viên, đại diện người dùng, chuyên gia bảo trì).

+ Mọi người sẽ đặt câu hỏi hoặc cho ý kiến bổ sung về một số lĩnh vực để bảo đảm chất lượng kỹ thuật của tài liệu hoặc sản phẩm.

+ Buổi giám định có thể xảy ra vào bất kỳ lúc nào và bất kỳ đâu trong việc phát triển sản phẩm phần mềm. Mục đích chính của họp giám định chỉ là để tìm lỗi nhanh, không tìm giải pháp. Sau giám định, tác giả của phải làm lại sửa mọi lỗi.

### **Tại sao trong nhóm walkthrough và inspection, luôn phải có đại diện của workflow tiếp theo?**

Bởi vì người đại diện cho nhóm workflow tiếp theo sẽ chuyển những thông tin công việc từ workflow trước về thông tin công việc của mình.

### **Tại sao trong nhóm walkthrough và inspection, luôn phải có đại diện của workflow hiện tại?**

Bởi vì người đại diện cho nhóm workflow hiện tại sẽ đại diện trình bày giải thích các lỗi và giải thích về các vấn đề hiện tại cho các thành viên khác trong nhóm, vì họ là người nắm rõ nhất các vấn đề và thông tin về workflow hiện tại. Khi có ý kiến phản biện hoặc thắc mắc, người này có thể giải đáp thắc mắc.

### **Tại sao nói inspection và walkthrough là hướng tài liệu, mà không phải hướng vào người tham gia?**

Các thành viên trong nhóm trước khi đến họp đều phải chuẩn bị bản tài liệu ghi các lỗi, thắc mắc của bản thân về dự án, sau đó tập họp lại để mọi người cùng xem xét. Hướng vào tài liệu vì cuộc họp tập trung vào tài liệu, tức là danh sách ghi lỗi chứ không hướng vào người tham gia tức là không quan tâm danh sách lỗi đó là của ai, của người có kinh nghiệm nhiều hay ít, của thành viên nhóm nào.

### **Tại sao người phụ trách nhóm inspection hay walkthrough phải là người của nhóm SQA?**

Các đại diện của workflow tham gia vào nhóm với vai trò như khách mời, không đủ trình độ, khả năng để nắm rõ vấn đề nhất như người của nhóm SQA, người nhóm SQA vai trò như chủ nhà và họ có trình độ chuyên môn cũng như có hiểu biết rõ về những gì đang diễn ra nhất.

### **So sánh walkthrough và inspection ?**

Team Inspection thường gồm 4 người. Vd cho Inspect design gồm có: Moderator, designer, implementer, tester.	Walkthrough: Gồm có 4-6 người, 1 người viết đặc tả (specification), 1 người quản lý workflow phân tích, đại diện khách hàng, 1
---	--



<ul style="list-style-type: none"> <li>• <b>Inspection:</b> Kỹ thuật đánh giá chính thức. Tài liệu, sản phẩm... được những người không phải là tác giả hoặc trực tiếp liên quan (Người kiểm duyệt, tác giả, tester, thiết kế, coder) kiểm tra một cách chi tiết để phát hiện lỗi, các vi phạm tiêu chuẩn, hoặc các vấn đề khác (nếu có).</li> </ul> <p>+ Về cơ bản, nó được tổ chức và thực hiện chặt chẽ hơn walkthrough. Vai trò của những người tham gia được phân định rõ ràng. Tài liệu chuẩn bị cho việc xem xét được chuẩn bị trước chu đáo.</p> <p>+ Quá trình duyệt thảo bắt đầu sau giai đoạn code và unit test. Sau buổi họp các lỗi tìm đc sẽ đc sửa lại, rồi đem ra duyệt thảo lại cho đến khi đạt tiêu chuẩn mới kết thúc quá trình này.</p>	<p>người thuộc team thiết kế, 1 đại diện thuộc SQA (leader).</p> <ul style="list-style-type: none"> <li>• <b>Walkthrough:</b> Kỹ thuật đánh giá không chính thức (nên ko có ng quản lý, giám đốc dự án). Những người tham gia phải xem tài liệu trước cuộc họp (ít nhất vài ngày). Tác giả giải thích tài liệu/ sản phẩm đó cho nhóm (tác giả, điều phối viên, giám định viên, đại diện người dùng, chuyên gia bảo trì).</li> </ul> <p>+ Mọi người sẽ đặt câu hỏi hoặc cho ý kiến bổ sung về một số lĩnh vực để bảo đảm chất lượng kỹ thuật của tài liệu hoặc sản phẩm.</p> <p>+ Buổi giám định có thể xảy ra vào bất kì lúc nào và bất kì đâu trong việc phát triển sản phẩm phần mềm. Mục đích chính của họp giám định chỉ là để tìm lỗi nhanh, ko tìm giải pháp. Sau giám định, tác giả của phải làm lại sửa mọi lỗi.</p>
--	--

### Bổ sung

Inspection	Walkthrough
Là kỹ thuật đánh giá chính thức	Là kỹ thuật đánh giá không chính thức
Do nhóm dự án khởi xướng	Được khởi xướng bởi tác giả
Một nhóm những người có liên quan từ các bộ phận khác nhau tham gia vào cuộc kiểm tra	Thông thường các thành viên trong cùng một dự án sẽ tham gia vào hướng dẫn. Tác giả tự đóng vai trò lãnh đạo bước đi.
Danh sách kiểm tra được sử dụng để tìm lỗi.	Không có danh sách kiểm tra nào được sử dụng.
Các quy trình kiểm tra bao gồm tổng quan, chuẩn bị, kiểm tra, làm lại và theo dõi.	Quá trình hướng dẫn bao gồm tổng quan, ít hoặc không chuẩn bị, kiểm tra ít hoặc không chuẩn bị (cuộc họp hướng dẫn thực tế), làm lại và theo dõi.
Quy trình chính thức hóa trong từng bước.	Không có thủ tục chính thức trong các bước.
Việc kiểm tra mất nhiều thời gian hơn vì	Thời gian ngắn hơn dành cho việc hướng



danh sách các mục trong danh sách kiểm tra được theo dõi để hoàn thành.	dẫn vì không có danh sách kiểm tra chính thức nào được sử dụng để đánh giá chương trình.
Cuộc họp được lên kế hoạch với các vai trò cố định được giao cho tất cả các thành viên tham gia.	Không có kế hoạch
Đầu đọc đọc mã sản phẩm. Mọi người đều kiểm tra nó và đưa ra những phát hiện.	Tác giả đọc mã sản phẩm và đồng đội của anh ấy đưa ra các khiếm khuyết hoặc đề xuất.
Người ghi chép ghi lại các khiếm khuyết.	Tác giả ghi lại những khiếm khuyết và đề xuất của đồng đội.
Người điều hành có vai trò là người điều hành đảm bảo rằng các cuộc thảo luận diễn ra theo hướng hiệu quả.	Không chính thức, vì vậy không có người điều hành.

### **Phần mềm còn lỗi hay không khi thực hiện hết các ca kiểm thử được thiết kế? Khi nào dừng kiểm thử?**

Phần mềm sẽ còn lỗi khi thực hiện hết các ca kiểm thử được thiết kế. Việc kết thúc kiểm thử dừng lại khi:

- Ứng dụng phần mềm được đảm bảo đã hoàn thành thỏa đáng, đáp ứng yêu cầu khách hàng.
- Tất cả các BUG (lỗi) mà người kiểm thử nỗ lực tìm thấy đã được giải quyết.
- 100% các BUG nghiêm trọng mức 1,2 (high, medium) đã được sửa xong.
- Kế hoạch bàn giao sản phẩm đã sẵn sàng.
- Các tài liệu liên quan đến ứng dụng phần mềm làm ra đã hoàn thành và luôn được cập nhật.

### **Kiểm thử hệ thống nhằm kiểm tra cái gì? Ai thực hiện? Các phương pháp?**

- (Product Testing): Diễn ra sau khi quá trình kiểm tra tích hợp được hoàn tất. Mục đích bảo đảm phần mềm hoàn tất. Thực hiện bởi SQA Group. Được thực hiện trên 2 loại phần mềm Cost Of The Shelf software và Custom Software.

- Với COTS Software, sẽ có 2 giai đoạn là alpha và beta. nhiều khách hàng sẽ được chọn để thử nghiệm và gửi feedback lại.

- Với Custom Software, SQL Group phải bảo đảm rằng Acceptance Test (bảo đảm đúng theo đặc tả) phải hiệu quả. Nếu không developer sẽ bị mất uy tín.

- Kiểm tra bao gồm các phương pháp:

+ Theo mức độ chi tiết của các bộ phận hợp thành phần mềm:

Kiểm thử đơn vị (Unit);

Kiểm thử hệ thống (System);

Kiểm thử tích hợp (Integration).

+ Theo phương pháp kiểm thử:

Kiểm thử hộp đen: Kiểm thử chức năng;

Kiểm thử hộp trắng: Kiểm thử cấu trúc.

### **Trình bày các kiểm thử được thực hiện trong kiểm thử hệ thống?**

Bước 1: Lên plan test.

Bước 2: Phân tích và thiết kế (Tạo testcase và các bước kiểm tra chi tiết cho mỗi version).

Bước 3: Thực thi test bao gồm thực hiện test và chạy test( chuẩn bị data test, chạy case và so sánh kết quả).

Bước 4: Đánh giá kết quả thực thi và báo cáo kết quả test.

Bước 5: Đóng hoạt động kiểm thử

### **Kiểm thử chấp nhận là gì? Trong đó có những kiểm thử nào được thực hiện? Phân biệt.**

- **Acceptance Test** (hay Acceptance Testing) dịch ra tiếng Việt là: **Kiểm thử chấp nhận**. Đây là 1 kiểm thử được thực hiện nhằm **xác định xem hệ thống phần mềm có đạt yêu cầu kỹ thuật hay không**. Nhờ việc **kiểm tra các hành vi của hệ thống** qua những dữ liệu thực tế, việc kiểm thử chấp nhận giúp **xác định xem hệ thống có đáp ứng được các tiêu chí và yêu cầu của khách hàng hay không**.
- Quy trình thực hiện:
  - Phân tích những yêu cầu nghiệp vụ của phần mềm;
  - Tạo kế hoạch Acceptance Test;
  - Xác định mọi kịch bản kiểm thử;
  - Tạo ra các trường hợp kiểm thử chấp nhận;

- Chuẩn bị data test cần đảm bảo giống với data thật nhất có thể;
- Thực hiện kiểm thử chấp nhận;
- Ghi nhận kết quả cuối cùng;
- Xác nhận các chức năng sản phẩm.

Phân biệt:

- + Alpha & Beta Testing thường diễn ra trong môi trường phát triển, thực hiện bởi nhân lực nội bộ. Số ít người dùng tiềm năng cũng có thể thực hiện Alpha Testing với điều kiện nó được diễn ra trong môi trường phát triển. Nhờ những vấn đề được thu thập từ Alpha & Beta Testing, bạn có thể xác định được một số lỗi tồn tại của sản phẩm để cải thiện chúng.
- + Contract Acceptance Test (dịch ra tiếng Việt là: Kiểm tra chấp nhận hợp đồng) được thực hiện để kiểm tra các tiêu chí và thông số kỹ thuật đã xác định trong hợp đồng. Những tiêu chí và thông số có liên quan sẽ được nhóm dự án xác định và chấp nhận khi nhóm hoàn toàn đồng ý với hợp đồng.
- + Regulation Acceptance Testing (dịch ra tiếng Việt là: Kiểm tra chấp nhận quy định) được thực hiện nhằm mục đích kiểm tra xem phần mềm có tuân thủ các quy định không. Trong quá trình kiểm tra, Tester cần đặc biệt lưu ý tới những quy định của chính phủ và yêu cầu về mặt pháp lý.
- + Operational Acceptance Testing (dịch ra tiếng Việt là: Thử nghiệm sẵn sàng hoạt động). Nó giúp đảm bảo các quy trình thực hiện công việc để cho phép phần mềm hoặc hệ thống được sử dụng. Trong bước kiểm thử này sẽ gồm: Các quy trình công việc cho kế hoạch dự phòng -> quy trình đào tạo người dùng -> quy trình bảo trì & quy trình bảo mật.
- + Black Box Testing (dịch ra tiếng Việt là: Kiểm thử hộp đen). Đây là một phần của kiểm thử chấp nhận người dùng. Phương pháp kiểm thử này giúp phân tích các chức năng, không cho phép người kiểm tra thấy được các cấu trúc code bên trong phần mềm hoặc hệ thống. Để làm tốt Black Box Testing thì bạn cần biết về những yêu cầu mà phần mềm phải đáp ứng.

### **Test trường hợp sai kiểu dữ liệu đầu vào thì thuộc thể loại test gì?**

Thuộc loại Database Test Cases . Bởi nhóm thử nghiệm sẽ sử dụng các truy vấn SQL để phát triển các trường hợp kiểm thử dữ liệu. Kiểm tra cơ sở dữ liệu được sử dụng để xác minh nhà phát triển đã viết mã theo cách lưu trữ và xử lý dữ liệu một cách nhất quán, an toàn.