



Kiểm Định Chất Lượng Phần Mềm

Bộ môn : Công nghệ Phần mềm
Team : Software Testing

Kiểm Định Chất Lượng Phần Mềm



Bộ môn : Công nghệ Phần mềm
Team : Software Testing



KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM

3 BM, CNPM

Đánh giá môn học:

- **Tiểu luận: 30%**
 - Chuyên cần: 10%
 - Kiểm tra thường xuyên: 20%
- **Cuối khóa: 70%**
 - Bài tập nhóm: 20%
 - Đồ án + Báo cáo: 50%

KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM

3 BM, CNPM

Đánh giá môn học:

- **Tiểu luận: 30%**
 - Chuyên cần: 10%
 - Kiểm tra thường xuyên: 20%
- **Cuối khóa: 70%**
 - Bài tập nhóm: 20%
 - Đồ án + Báo cáo: 50%

- # KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM
- 3 BM, CNPM
- ## Đánh giá môn học:
- **Tiểu luận: 30%**
 - Chuyên cần: 10%
 - Kiểm tra thường xuyên: 20%
 - **Cuối khóa: 70%**
 - Bài tập nhóm: 20%
 - Đồ án + Báo cáo: 50%

KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM

4 BM, CNPM

Tài liệu tham khảo:

- [1]. Thạc Bình Cường, *Kiểm thử và đảm bảo chất lượng phần mềm*, Trường Đại học Bách Khoa Hà Nội, 2011.
- [2]. Phạm Ngọc Hùng, Trương Anh Hoàng, Đặng Văn Hưng, *Giáo trình kiểm thử phần mềm*, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội, 2014.

KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM

4 BM, CNPM

Tài liệu tham khảo:

- [1]. Thạc Bình Cường, *Kiểm thử và đảm bảo chất lượng phần mềm*, Trường Đại học Bách Khoa Hà Nội, 2011.
- [2]. Phạm Ngọc Hùng, Trương Anh Hoàng, Đặng Văn Hưng, *Giáo trình kiểm thử phần mềm*, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội, 2014.

- # KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM
- 4 BM, CNPM
- ## Tài liệu tham khảo:
- [1]. Thạc Bình Cường, *Kiểm thử và đảm bảo chất lượng phần mềm*, Trường Đại học Bách Khoa Hà Nội, 2011.
 - [2]. Phạm Ngọc Hùng, Trương Anh Hoàng, Đặng Văn Hưng, *Giáo trình kiểm thử phần mềm*, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội, 2014.

KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM

5 | BM, CNPM

Tài liệu tham khảo:

- [3]. Cem Kaner, Jack Falk, Hung Q. Nguyen: *Testing Computer Software*, 2nd edition. Canada, Wiley, 1999.
- [4]. Hung Q. Nguyen: *Testing applications on the web*. USA, Wiley, 2003.
- [5]. Paul Ammann, Jeff Offutt: *Introduction to Software Testing*, Cambridge University Press, 2008.
- [6]. Glenford J. Myers: *The art of Software Testing*, John Wiley & Son, 2004.
- [7]. LogiGear: *Basic Software Testing Skills*, LogiGear Corporation, 2009.

KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM

5 | BM, CNPM

Tài liệu tham khảo:

- [3]. Cem Kaner, Jack Falk, Hung Q. Nguyen: *Testing Computer Software*, 2nd edition. Canada, Wiley, 1999.
- [4]. Hung Q. Nguyen: *Testing applications on the web*. USA, Wiley, 2003.
- [5]. Paul Ammann, Jeff Offutt: *Introduction to Software Testing*, Cambridge University Press, 2008.
- [6]. Glenford J. Myers: *The art of Software Testing*, John Wiley & Son, 2004.
- [7]. LogiGear: *Basic Software Testing Skills*, LogiGear Corporation, 2009.

- # KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM
- 5 | BM, CNPM
- ## Tài liệu tham khảo:
- [3]. Cem Kaner, Jack Falk, Hung Q. Nguyen: *Testing Computer Software*, 2nd edition. Canada, Wiley, 1999.
 - [4]. Hung Q. Nguyen: *Testing applications on the web*. USA, Wiley, 2003.
 - [5]. Paul Ammann, Jeff Offutt: *Introduction to Software Testing*, Cambridge University Press, 2008.
 - [6]. Glenford J. Myers: *The art of Software Testing*, John Wiley & Son, 2004.
 - [7]. LogiGear: *Basic Software Testing Skills*, LogiGear Corporation, 2009.

KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM

6 BM, CNPM

Mục tiêu môn học:

- Giới thiệu về nghề Tester/QA/QC
- Cung cấp kiến thức và kỹ thuật cơ bản trong lĩnh vực kiểm định chất lượng phần mềm: khái niệm cơ bản; quy trình; các loại kiểm thử; kế hoạch kiểm thử; thiết kế testcase, testdata, ...
- Làm quen các công cụ hỗ trợ kiểm thử phần mềm: Junit & Nunit; Selenium, UI Automation, ...

KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM

6 BM, CNPM

Mục tiêu môn học:

- Giới thiệu về nghề Tester/QA/QC
- Cung cấp kiến thức và kỹ thuật cơ bản trong lĩnh vực kiểm định chất lượng phần mềm: khái niệm cơ bản; quy trình; các loại kiểm thử; kế hoạch kiểm thử; thiết kế testcase, testdata, ...
- Làm quen các công cụ hỗ trợ kiểm thử phần mềm: Junit & Nunit; Selenium, UI Automation, ...

- # KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM
- 6 BM, CNPM
- ## Mục tiêu môn học:
- Giới thiệu về nghề Tester/QA/QC
 - Cung cấp kiến thức và kỹ thuật cơ bản trong lĩnh vực kiểm định chất lượng phần mềm: khái niệm cơ bản; quy trình; các loại kiểm thử; kế hoạch kiểm thử; thiết kế testcase, testdata, ...
 - Làm quen các công cụ hỗ trợ kiểm thử phần mềm: Junit & Nunit; Selenium, UI Automation, ...

KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM

7 BM, CNPM

Kết quả kỳ vọng:

Kiến thức

- Hiểu vai trò kiểm thử trong quy trình.
- Hiểu quy trình, các giai đoạn kiểm thử.
- Hiểu các kỹ thuật kiểm thử khác nhau.

Kỹ năng

- Nhận diện, phân tích và thiết kế kịch bản kiểm thử.
- Thực hiện kịch bản kiểm thử, đánh giá kết quả.
- Thực hiện phân tích, kiểm thử một ứng dụng theo nhóm.

Thái độ

- Có khả năng làm việc nhóm.
- Có tính kỷ luật, siêng năng, cẩn thận, tỉ mỉ.

KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM

8 BM, CNPM

Giới thiệu về nghề Tester



GIỚI THIỆU VỀ NGHỀ TESTER

9 BM, CNPM

Nghề tester là gì?

Bất kỳ một phần mềm hay ứng dụng nào trước khi đưa vào hoạt động đều phải trải qua **khâu kiểm tra**. Những người phụ trách công việc này được gọi là **Tester - Chuyên viên kiểm thử phần mềm**.

GIỚI THIỆU VỀ NGHỀ TESTER

10 BM, CNPM

Nghề tester làm gì?

- **Tìm kiếm các lỗi** của hệ thống phần mềm.
- **Thẩm định, xác minh** xem hệ thống phần mềm có đáp ứng các yêu cầu kỹ thuật và yêu cầu nghiệp vụ hay không.

GIỚI THIỆU VỀ NGHỀ TESTER

11 BM, CNPM

Tìm năng của nghề tester?

- Được nhiều bạn nữ lựa chọn.
- Có cơ hội việc làm cao.

KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM

12 BM, CNPM

Những tố chất để làm tốt công việc tester:

- Tester cần phải hiểu và **thông thạo** ít nhất một **ngôn ngữ lập trình**.
- Có được những kỹ năng **phân tích, thiết kế, lập trình**.
- **Hiểu** biết về **các ứng dụng** khác nhau của các phần mềm.
- Tester cũng cần có **trình độ tiếng Anh** để đọc, hiểu, viết được tài liệu chuyên ngành, để tiếp cận kiến thức mới của thế giới.
- "Một kỹ sư kiểm tra chất lượng vừa phải có **cái nhìn của người phát triển phần mềm**, vừa phải đứng ở góc độ là **"người dùng đầu cuối"**."

Chương 1: CÁC KHÁI NIỆM

I. Yêu cầu phần mềm (Software Requirement)

Yêu cầu (requirement):

Là một đặc trưng của hệ thống, hay là sự mô tả những việc mà hệ thống có khả năng thực hiện để hoàn thành mục tiêu của hệ thống.

Phân loại Yêu cầu

❖ Yêu cầu chức năng (Functional Requirement)

- Là các yêu cầu mô tả chức năng của một hệ thống phần mềm, mô tả phần mềm phải làm gì, có khả năng thực hiện những công việc gì.
- Một chức năng được mô tả thông qua các *dữ liệu đầu vào, cách thức xử lý và dữ liệu được kết xuất*.
- Một số yêu cầu chức năng như: tính toán, lưu trữ, tìm kiếm, kết xuất, backup, restore, đa người dùng,...

Phân loại Yêu cầu

Ví dụ: Trong hệ thống quản lý thư viện

Người dùng có thể tìm kiếm, download, in những bài báo.

Người dùng được cấp một vùng lưu trữ riêng để có thể copy, lưu trữ tài liệu lâu dài, ...

Phân loại Yêu cầu

❖ Yêu cầu phi chức năng (Non-Functional Requirement)

Là các yêu cầu mô tả về *tính dễ sử dụng, độ tin cậy, độ ổn định, tốc độ xử lý* của phần mềm, yêu cầu về *sự sẵn sàng hỗ trợ* trong quá trình triển khai đối với nhà cung cấp phần mềm.

Phân loại Yêu cầu

Một số yêu cầu phi chức năng:

- Độ tin cậy, thời gian đáp ứng, các yêu cầu về lưu trữ.
- Ngôn ngữ lập trình.
- Yêu cầu của người sử dụng: dễ sử dụng, thân thiện.
- Phù hợp với các chính sách của tổ chức sử dụng hệ thống.
- Yêu cầu tương thích giữa phần cứng và phần mềm...

Phân loại Yêu cầu

19 BM, CNPM

Ví dụ: Trong hệ thống quản lý thư viện

- **Yêu cầu sản phẩm:**
Giao diện người dùng không chứa applet java.
- **Yêu cầu cơ sở dữ liệu quản lý:**
SQL Server 2008 trở lên.
- **Yêu cầu ngoài:**
Hệ thống không được lộ thông tin của khách hàng (tên, số tham chiếu...)

Các nguyên nhân gây ra lỗi phần mềm

20 BM, CNPM

1. Định nghĩa sai yêu cầu của khách hàng:

- Đây được coi là gốc rễ của việc gây ra lỗi phần mềm.
- Hiểu sai yêu cầu của khách hàng.
- Yêu cầu của khách hàng không được làm rõ.
- Triển khai phần mềm thiếu yêu cầu của khách hàng.
- Khách hàng đưa ra quá nhiều yêu cầu không cần thiết và không liên quan.

Các nguyên nhân gây ra lỗi phần mềm

21 BM, CNPM

2. Thất bại trong việc giao tiếp giữa người phát triển và khách hàng

- Có sự không hiểu cấu trúc của tài liệu yêu cầu phần mềm.
- Không nắm bắt được những thay đổi được viết trong tài liệu yêu cầu.
- Những thay đổi được yêu cầu từ khách hàng nhưng không được lưu dưới dạng văn bản.
- Thiếu sự chú ý tới:
 - Thông điệp của khách hàng đề cập tới việc thay đổi yêu cầu.
 - Trả lời của khách hàng tới những câu hỏi mà developer đặt ra.

Các nguyên nhân gây ra lỗi phần mềm

22 BM, CNPM

3. Tạo ra độ lệch cố ý trong yêu cầu phần mềm

- Lập trình viên sử dụng những module phần mềm có sẵn từ những dự án trước mà không thay đổi cho phù hợp với yêu cầu của dự án mới nhằm tiết kiệm thời gian.
- Bỏ qua một vài yêu cầu của phần mềm do thời gian quá gấp hoặc chi phí không đủ đáp ứng.

Các nguyên nhân gây ra lỗi phần mềm

23 BM, CNPM

4. Lỗi mã hóa

- Lỗi logic.
- Lỗi cú pháp.
- Lỗi thời gian chạy.

5. Không tuân theo các tài liệu và cấu trúc code

- Không tuân theo các chuẩn tài liệu (templates...).
- Không tuân theo các cấu trúc mã hóa.

Các nguyên nhân gây ra lỗi phần mềm

24 BM, CNPM

6. Rút ngắn quá trình kiểm thử phần mềm

- Do áp lực về thời gian, tiến độ hoàn thành dự án.
- Lập kế hoạch kiểm thử không đầy đủ.
- Không báo cáo đầy đủ các lỗi.
- Báo cáo không chính xác lỗi.

Các nguyên nhân gây ra lỗi phần mềm

25 BM, CNPM

7. Lỗi thủ tục

- Chỉ dẫn cho người dùng những hoạt động cần thiết ở một quá trình.
- Nó quan trọng trong các hệ thống phần mềm phức tạp khi quá trình xử lý được thực hiện qua nhiều bước.
- Mỗi bước có nhiều dạng dữ liệu và cho phép kiểm tra kết quả trung gian.

Các nguyên nhân gây ra lỗi phần mềm

26 BM, CNPM

8. Lỗi tài liệu

- Sai sót trong hồ sơ thiết kế.
- Sai sót trong việc lập tài liệu hướng dẫn sử dụng.
- Các danh sách chức năng không có trong phần mềm nhưng lại có trong tài liệu.

II. Các khái niệm

27 BM, CNPM

Kiểm định (Tester):

Là người kiểm thử phần mềm để tìm kiếm các lỗi, sai sót, hay bất cứ vấn đề nào mà có thể ảnh hưởng đến chất lượng phần mềm.

Có nhiệm vụ **khảo sát, chạy thử** để bảo đảm phần mềm **thỏa mãn các yêu cầu** về chức năng và khả năng vận hành mà nó phải có, **báo cáo các lỗi** nếu có để các bộ phận liên quan chỉnh sửa. Công việc của kỹ sư kiểm định liên quan đến sản phẩm (*product*).

II. Các khái niệm

28 BM, CNPM

Đảm bảo chất lượng (Quality Assurance - QA):

Là người chịu trách nhiệm đảm bảo chất lượng sản phẩm thông qua việc đưa ra quy trình làm việc giữa các bên liên quan.

Có nhiệm vụ **giám sát** để bảo đảm các tiêu chuẩn và quy trình sản xuất phần mềm được định nghĩa và tuân thủ nghiêm túc, hướng đến mục tiêu các sản phẩm trung gian cũng như sản phẩm sau cùng của **dự án** nhằm **thỏa mãn các tiêu chuẩn và yêu cầu** đã định trước đó. Công việc của kỹ sư đảm bảo chất lượng liên quan đến quy trình (*process*).

II. Các khái niệm

29 BM, CNPM

Kiểm soát chất lượng (Quality Control - QC):

Là người chịu trách nhiệm thực hiện công việc **kiểm soát, kiểm tra, đánh giá chất lượng phần mềm**.

Có nhiệm vụ kiểm tra chất lượng của phần mềm, đảm bảo chất lượng của phần mềm khi đưa ra thị trường sử dụng.



II. Các khái niệm

30 BM, CNPM

Ví dụ:

- Kiểm tra để bảo đảm các giải thuật khi viết code phải được chú thích rõ ràng,
- Các Yêu cầu khách hàng được xem xét cẩn thận và mọi người hiểu giống nhau,
- Các tài liệu đi kèm sản phẩm phải được kiểm tra trước khi gửi cho khách hàng.

II. Các khái niệm

31 BM, CNPM

Các yếu tố ảnh hưởng đến chất lượng

Có ba yếu tố ảnh hưởng tới chất lượng phần mềm (tam giác chất lượng)

- Phạm vi
- Thời gian
- Chi phí



II. Các khái niệm

32 BM, CNPM

Theo Glenford Myers:

- Kiểm thử (*testing*) là quá trình vận hành chương trình để nhằm mục đích tìm ra lỗi.

II. Các khái niệm

33 BM, CNPM

Theo IEEE, Kiểm thử là:

(1) Là quá trình vận hành hệ thống hoặc thành phần dưới những điều kiện xác định, quan sát hoặc ghi nhận kết quả và đưa ra đánh giá về hệ thống hoặc thành phần đó.

(2) Là quá trình phân tích phần mềm để tìm ra sự khác biệt giữa điều kiện thực tế và điều kiện yêu cầu và dựa vào điểm khác biệt đó để đánh giá tính năng phần mềm.

II. Các khái niệm

34 BM, CNPM

The Art of Software Testing

Là hoạt động kiểm tra xem phần mềm có chạy chính xác hay không (*Verification*) và có thỏa mãn yêu cầu của khách hàng hay không (*Validation*) nhằm hướng tới mục tiêu Chất lượng cho phần mềm.

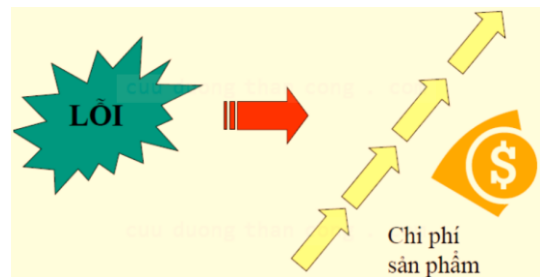
Mục tiêu của kiểm thử

35 BM, CNPM

- Tìm ra được càng nhiều lỗi càng tốt trong điều kiện về thời gian đã định và nguồn lực sẵn có.
- Chứng minh rằng sản phẩm phần mềm phù hợp với các đặc tả của nó.
- Xác thực chất lượng kiểm thử phần mềm đã dùng chi phí và nỗ lực tối thiểu.
- Thiết kế tài liệu kiểm thử một cách có hệ thống và thực hiện nó sao cho có hiệu quả, tiết kiệm được thời gian công sức.

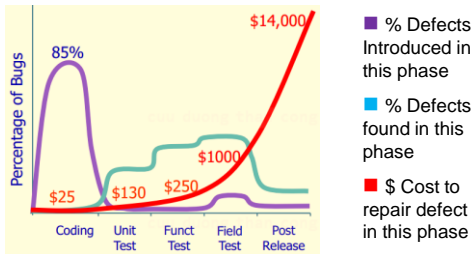
Tâm quan trọng của kiểm thử

36 BM, CNPM



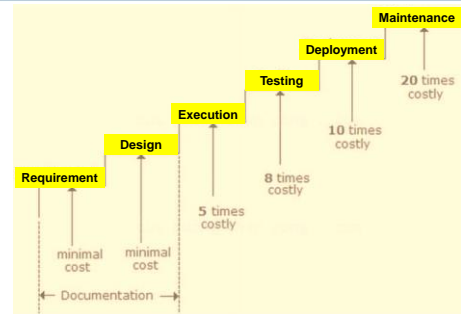
Tâm quan trọng của kiểm thử

37 BM, CNPM



Tâm quan trọng của kiểm thử

38 BM, CNPM



Vai trò của kiểm thử

39 BM, CNPM

Vai trò kiểm thử trong suốt quy trình sống của phần mềm:

- Kiểm thử không tồn tại độc lập.
- Các hoạt động của kiểm thử luôn gắn liền với các hoạt động phát triển phần mềm.
- Các mô hình phát triển phần mềm khác nhau cần các cách tiếp cận kiểm thử khác nhau.

Tâm quan trọng của kiểm thử

40 BM, CNPM

Tại sao cần thực hiện kiểm thử?

- Để xem xét chất lượng sản phẩm.
- Để phát hiện ra lỗi.
- Ví dụ:** Khách hàng có thể rút tiền ở máy ATM với số tiền tối đa là 250\$/1 giao dịch.
 - Người kiểm thử 1:** Thử 3 lần với 3 yêu cầu: 50\$, 150\$, 250\$ thấy máy đều nhả ra số tiền chính xác, kết luận chức năng rút tiền hoạt động đúng yêu cầu của khách hàng là yêu cầu rút ra bao nhiêu đều trả về đúng bấy nhiêu tiền.
 - Người kiểm thử 2:** Yêu cầu số tiền là 300\$, máy vẫn nhả ra đúng 300\$ mà không đưa ra thông báo số tiền rút bị quá hạn, như vậy là có lỗi mà người kiểm thử 1 không tìm ra được.

Các nguyên tắc của kiểm thử

41 BM, CNPM

Các nguyên tắc của kiểm thử

42 BM, CNPM

Trong kiểm thử có 7 nguyên tắc cơ bản:

- Kiểm thử chứng minh sự hiện diện của lỗi.
- Kiểm thử toàn bộ là không thể
- Kiểm thử càng sớm càng tốt.
- Lỗi thường được phân bố tập trung (**Defect Clustering**).
- Nghịch lý thuốc trừ sâu (**Pesticide paradox**).
- Kiểm thử phụ thuộc vào ngữ cảnh.
- Quan niệm sai lầm về việc "hết lỗi".

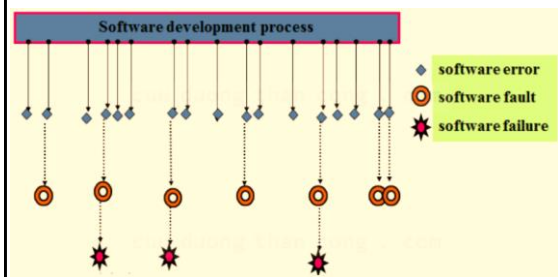
II. Các khái niệm

43 BM, CNPM

- **Lỗi (Error):** Là các lỗi làm do con người gây ra.
- **Sai sót (Fault):** Sai sót gây ra lỗi (gồm dư thừa, bỏ sót).
- **Hỏng hóc (Failure):** Xảy ra khi sai sót được thực thi.
- **Kết quả không mong đợi, hậu quả (Incident):** Là những kết quả do sai sót gây ra.
- **Trường hợp thử (Test case):** Một trường hợp thử bao một tập các giá trị đầu vào và một danh sách các kết quả đầu ra mong muốn.

II. Các khái niệm

44 BM, CNPM



II. Các khái niệm

45 BM, CNPM

Xác minh (Verification):

- Xác minh là quy trình xác định xem sản phẩm của một công đoạn trong quy trình phát triển phần mềm có thỏa mãn các yêu cầu đặt ra trong công đoạn trước hay không? (*Ta có đang xây dựng đúng sản phẩm mà được đặc tả không?*).
- Xác minh quan tâm tới việc ngăn chặn lỗi giữa các công đoạn với nhau.
- Xác minh thường là hoạt động kỹ thuật và nó có sử dụng các kiến thức về các yêu cầu, các đặc tả rời rạc của phần mềm.
- Các hoạt động của xác minh bao gồm: Kiểm thử (*Testing*) và Rà soát lại (*Review*).

II. Các khái niệm

46 BM, CNPM

Thẩm định (Validation):

- Thẩm định là tiến trình nhằm chỉ ra toàn bộ hệ thống đã phát triển xong phù hợp với tài liệu mô tả yêu cầu. Thẩm định là quá trình kiểm chứng chúng ta xây dựng phần mềm có đúng theo yêu cầu khách hàng không?
- Thẩm định chỉ quan tâm đến sản phẩm cuối cùng không còn lỗi.

II. Các khái niệm

47 BM, CNPM

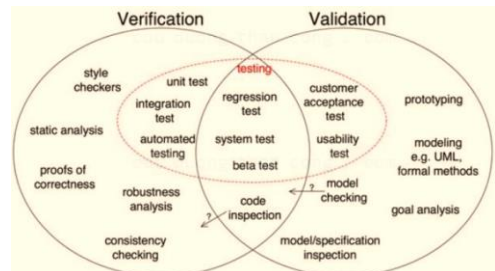
So sánh giữa Xác minh và Thẩm định:

- **Xác minh (Verification):** quan tâm đến việc ngăn chặn lỗi giữa các công đoạn.
- **Thẩm định (Validation):** quan tâm đến sản phẩm cuối cùng không còn lỗi.

II. Các khái niệm

48 BM, CNPM

So sánh giữa Xác minh và Thẩm định:



II. Các khái niệm

49 BM, CNPM

- **Dữ liệu kiểm thử (test data):** Dữ liệu cần cung cấp để phần mềm có thể thực thi để kiểm thử.
- **Kịch bản kiểm thử (test scenario):** Các bước thực hiện khi kiểm thử.
- **Kỹ sư kiểm thử (tester):** Người thực hiện kiểm thử.

II. Các khái niệm

50 BM, CNPM

- **Ca kiểm thử (test case):** chứa các thông tin cần thiết để kiểm thử thành phần phần mềm theo 1 mục tiêu xác định.
- **Test case gồm bộ 3 thông tin { tập dữ liệu đầu vào, thứ tự thực hiện, tập kết quả kỳ vọng }**
 - **Tập dữ liệu đầu vào (input):** gồm các giá trị dữ liệu cần thiết để thành phần phần mềm dùng và xử lý.

II. Các khái niệm

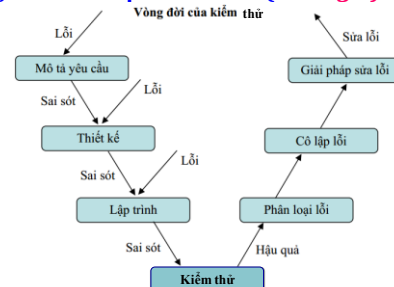
51 BM, CNPM

- **Tập kết quả kỳ vọng (output):** kết quả mong muốn sau khi thành phần phần mềm xử lý dữ liệu nhập.
- **Thứ tự thực hiện:** các bước để hoàn thành ca kiểm thử từ lúc nhập dữ liệu đầu vào tới lúc nhận được kết quả đã qua xử lý của phần mềm.

II. Các khái niệm

52 BM, CNPM

Vòng đời của việc kiểm thử (testing life cycle)



III. Phân loại kiểm thử

53 BM, CNPM

Phân loại kiểm thử theo mức độ chi tiết của các bộ phận hợp thành phần mềm:

- Mức kiểm thử đơn vị (Unit/Module)
- Mức kiểm thử tích hợp (Integration)
- Mức kiểm thử hệ thống (System)
- Mức kiểm thử chấp nhận (Acceptance)

III. Phân loại kiểm thử

54 BM, CNPM

Phân loại dựa vào mục đích kiểm thử:

- Kiểm thử cấu hình (Configuration testing)
- Kiểm thử sơ lược (Smoke testing)
- Kiểm thử chức năng (Functional testing)
- Kiểm thử hồi quy (Regression testing)
- Kiểm thử tải dữ liệu (Load testing)
- Kiểm thử mức độ đáp ứng (Stress testing)
- Kiểm thử hiệu suất (Performance testing)
- Kiểm thử bảo mật (Security testing)
- ...

III. Phân loại kiểm thử

65 BM, CNPM

Phân loại dựa vào kỹ thuật kiểm thử:

- **Kiểm thử hộp trắng (White box testing):** kiểm thử cấu trúc.
 - Kiểm thử theo góc nhìn thực hiện.
 - Cần có kiến thức về chi tiết thiết kế và thực hiện bên trong.
 - Kiểm thử dựa vào phủ các lệnh, các nhánh, phủ các điều kiện con.

III. Phân loại kiểm thử

66 BM, CNPM

Phân loại dựa vào kỹ thuật kiểm thử:

- **Kiểm thử hộp đen (Black box testing):** kiểm thử chức năng.
 - Kiểm thử theo góc nhìn sử dụng.
 - Kiểm thử dựa trên các yêu cầu và đặc tả sử dụng thành phần phần mềm.
 - Không đòi hỏi kiến thức về chi tiết thiết kế và thực hiện ở bên trong chương trình.

III. Phân loại kiểm thử

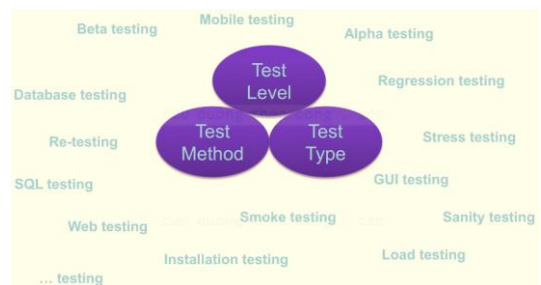
67 BM, CNPM

Phân loại dựa vào chiến lược kiểm thử:

- **Kiểm thử thủ công (Manually Test):**
 - Thực hiện kiểm thử mọi thứ bằng tay, từ viết test case đến thực hiện test.
- **Kiểm thử tự động (Automatically Test):**
 - Thực hiện một cách tự động các bước trong kịch bản kiểm thử bằng cách dùng một công cụ trợ giúp.
 - Kiểm thử tự động nhằm tiết kiệm thời gian kiểm thử.

Phương pháp kiểm thử

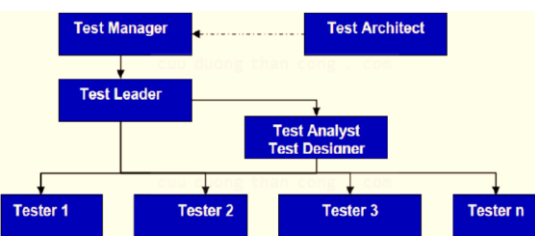
68 BM, CNPM



Đối tượng thực hiện kiểm thử

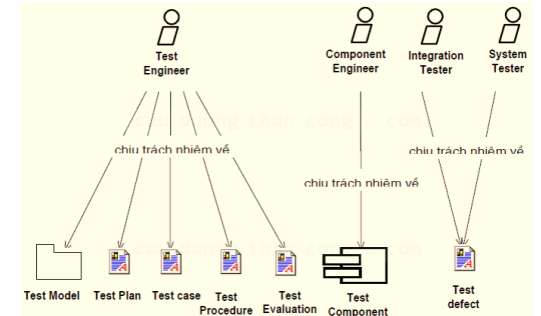
69 BM, CNPM

Sơ đồ tổ chức của đội kiểm thử



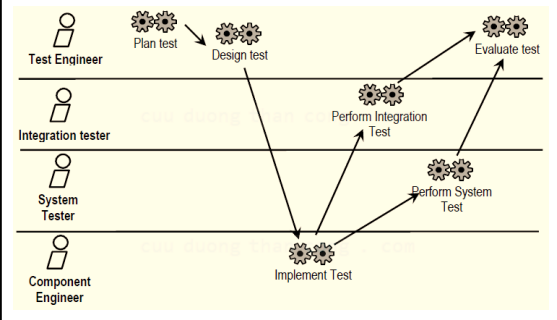
Các Worker và quy trình

70 BM, CNPM



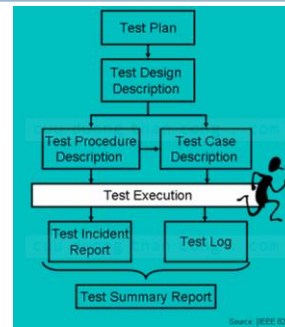
Các Worker và quy trình

61 BM, CNPM



Các Worker và quy trình

62 BM, CNPM



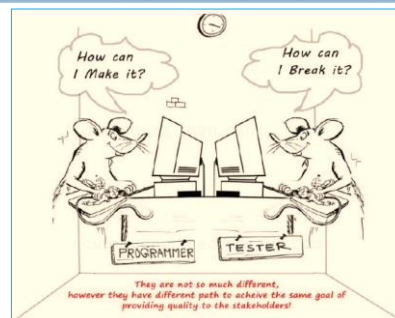
Các Worker và quy trình

63 BM, CNPM



Các Worker và quy trình

64 BM, CNPM



Các kỹ thuật kiểm thử

65 BM, CNPM

- Test tĩnh (Static Verification)**
 - Thực hiện kiểm chứng mà không cần thực thi chương trình.
 - Kiểm tra tính đúng đắn của các tài liệu có liên quan được tạo ra trong quá trình xây dựng ứng dụng.
 - Đạt được sự nhất quán và hiểu rõ hơn về hệ thống.
 - Giảm thời gian lập trình, thời gian và chi phí test,...
- Test động (Dynamic Testing)**
 - Thực hiện kiểm thử dựa trên việc thực thi chương trình.

Kiểm thử hộp trắng (white-box testing) - Còn gọi là Kiểm thử cấu trúc:

66 BM, CNPM

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.

Kiểm thử hộp đen (black-box testing) - Còn gọi là Kiểm thử chức năng

67 BM, CNPM

- 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.
- 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.

Vấn đề Kiểm thử tại biên

68 BM, CNPM

- 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.

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

Các loại Kiểm thử tầm rộng

69 BM, CNPM

- Kiểm thử bộ phận (Unit/Module testing):** Kiểm thử 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.
- Kiểm thử chấp nhận (Acceptance testing):** thực hiện bởi khách hàng.

Các loại Kiểm thử tầm rộng

70 BM, CNPM

Kiểm thử bộ phận (Unit/Module testing)

- Mục đích:** Xác minh module đưa ra đã được xây dựng đúng hay chưa?
- Vấn đề đặt ra:** giả sử module *I* sử dụng các module *H*, *K*. Nhưng các module *H* và *K* chưa sẵn sàng. Vậy cách nào để kiểm tra module *I* một cách độc lập?
- Giải pháp đề ra:** là *giả lập môi trường* của module *H* và *K*.

Các loại Kiểm thử tầm rộng

71 BM, CNPM

Kiểm thử tích hợp (Integration testing)

- Mục đích:** Là cách Kiểm thử bằng cách tích hợp vào hệ thống từng module một và kiểm tra.
- Ưu điểm:**
 - ✓ Dễ dàng tìm ra các lỗi vào ngay giai đoạn đầu.
 - ✓ Dễ dàng khoanh vùng các lỗi (*tích hợp n modules, sau đó $n + 1$ modules, ...*).

Các loại Kiểm thử tầm rộng

72 BM, CNPM

Kiểm thử hệ thống (System testing)

- Mục đích:** Đảm bảo toàn bộ hệ thống hoạt động như mong muốn của khách hàng.
 - Kiểm thử chức năng (*Function testing*)
 - Kiểm thử hiệu suất (*Performance testing*)
 - Kiểm thử mức độ đáp ứng (*Stress testing*)

Các loại Kiểm thử tầm rộng

73 BM, CNPM

Kiểm thử hệ thống (*System testing*)

- Kiểm thử cấu hình (*Configuration testing*)
- Kiểm thử ổn định (*Robustness testing*)
- Kiểm thử hồi phục (*Recovery testing*)
- Kiểm thử quá tải (*Overload testing*)
- Kiểm thử chất lượng (*Quality testing*)
- Kiểm thử cài đặt (*Installation testing*)

Các loại Kiểm thử tầm rộng

74 BM, CNPM

Kiểm thử chấp nhận (*Acceptance testing*)

- **Mục đích:** 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.

KIỂM ĐỊNH CHẤT LƯỢNG PHẦN MỀM

75 BM, CNPM

Thank for your attention!