**PHẦN 1: Các khái niệm căn bản về kiểm thử phần mềm**

**Bài 1: Kiểm thử phầm mềm là gì?**

**Các cách nhìn khác nhau về chất lượng (quality) của một sản phẩm phần mềm**

* Trước khi nói về kiểm thử phần mềm, chúng ta phải nói về chất lượng sản phẩm phần mềm:
  + Đứng ở các góc nhìn khác nhau, định nghĩa về một sản phẩm phần mềm chất lượng cũng sẽ khác nhau
    - Từ phía người dùng: sản phẩm phù hợp với mục đích sử dụng của người đó
    - Từ phía nhà phân bổ, cung cấp sản phẩm (product view): một sản phẩm đạt chất lượng là sản phẩm có được các đặc tính về sản phẩm, đạt được những tiêu chú đánh giá do nhà cung cấp sản phẩm đề ra
      * Tnhs chính xác
      * Tính bảo mật
      * Tính dễ sử dụng
      * …
    - Từ phía nhà sản suất ra sản phẩm phân mềm: đạt được chất lượng cần thiết để đưa ra dùng hay sử dụng phải được xây dựng đúng như những yêu cầu đặc tả (do người sử dụng hoặc nhà cung cấp sản phẩm đề ra)
  + Từ những góc nhìn trên, tam giác chất lượng của sản phẩm phần mềm sẽ được xây dựng như sau:
    - Đỉnh 1: view của nhà cung cấp sản phẩm (Product’s view)
    - Đỉnh 2: view của người sử dụng (user’s view)
    - Đỉnh 3: view của nhà sản xuất (manufacture’s view)
* Một sản phẩm phần mềm sẽ được xây dựng dựa trên yêu ầu người dùng + yêu cầu đặc tả
  + Thông thường, trong quy trình sản xuất phần mềm, nhà cung cấp sản phẩm phần mềm sẽ theo những yêu cầu của user để viết ra một customer requirement (hay có lúc được gọi là high level requirement)
  + Sau đó nhà sản xuất sẽ phát triển thành một tài liệu là requirement specification (yêu cầu đặc tả) mô tả chi tiết yêu cầu về các chức năng, tính năng của sản phẩm 🡪 Developers sẽ dựa vào đó để xẩy dựng / phát triển sản phẩm phần mềm
  + Khoảng cách giữa các đỉnh của tam giác giữa sản phẩm thực tế và yêu cầu chức năng, tính năng của sản phẩm càng lớn, nghĩa là **chất lượng (quality)** của sản phẩm càng kém và ngược lại.

**Kiểm thử phần mềm là gì?**

* Định nghĩa:
  + Theo IEEE 829: Testing là một quá trình thực hiện đánh giá một hệ thống, một sản phẩm tự động bằng tay để xác nhận rằng sản phẩm có thỏa mãn đặc tả hay không
  + Theo ISTQB: Testing là một quá trìn bao gồm một vòng lặp các hoạt động cả tĩnh và động, từ khi lập kế hoạch, chuẩn bị và đánh giá sản phẩm để xác định sản phẩm có thõa mãn yêu cầu đặc tả không, chứng minh sản phẩm đó có phù hợp với mục đích sử dụng không và để phát hiện lỗi.
* Mục tiêu cyả kiểm thử phần mềm:
  + Xác định sản phẩm có thõa mãn yêu cầu đặc tả hay không
  + Xác định sản phẩm có thõa mãn những cái người dùng cần không
  + Có được tự tin để cung cấp một sản phẩm có chất lượng không
  + Xác định các lỗi, nguyên nhân để đưa ra các hành động để hạn chế, ngăn chặn các lỗi trong tương lai
  + Có thể đưa ra các cải tiến quy trình, tìm thêm các công cụ để phục vụ công việc testing hiệu quả hơn

**Sự khác nhau giữa lý thuyết và thực tế:**

1. Testing là một công việc đơn giản, dễ làm: thực tế, testing là 1 công việc thách thức, có những trường hợp kiểm thử đòi hỏi kỹ năng phân tích cao
2. Ai cũng có thể làm manual testing: thực tế testing yêu cầu nhiều kỹ năng cần có và phải có để làm nghề tester
3. Testing để đảm bảo phần mềm sau khi kiểm thử không có lỗi, nhưng thức tế không phải lúc nào việc testing cũng có thể phát hiện được tất cả các lỗi của phần mềm
4. Test tự động sẽ tốt hơn, hiệu quả hơn việc test bằng tay: không thể tự động test hoàn toàn tất cả các trường hợp của phần mềm
5. Testing và debugging là giống nhau: thực tế là hoàn toàn khác nhau:
   1. Testing là tìm lỗi, debugging là hành động để sửa lỗi
   2. Testing thực hiện bởi tester, còn debugging thực hiện bởi developers
   3. Testing là để tìm được càng nhiều lỗi càng tốt, còn debugging là để loại bỏ những lỗi đó

1 B

2 D

3 D

4 D

5 A

**--------------- --------------- --------------- --------------- --------------- ---------------**

**Bài 2: Tại sao phải thực hiện kiểm thử phần mềm?**

**Tại sao việc kiểm thử phần mềm lại cần thiết**

1. Sản phẩm phần mềm nào cũng chắc chắn có lỗi, do đó sẽ luôn cần một người, nhóm người hoặc tổ chức độc lập đứng ra kiểm thử xem sản phẩm có lỗi hay không
2. Kiểm thử giúp tìm hiểu độ tin cậy của phần mềm
3. Các lỗi trong quá trình phần mềm được đưa vào sử dụng có thể rất tốn chi phí để khắc phụ và có thể gây ra những hậu quả nghiêm trọng
4. Tránh được sự kiện tụng của khách hàng
5. Doanh nghiệp muốn phát triển trong một lĩnh vực nào đó luôn cần đảm bảo sản phẩm do mình cung cấp đến người dùng có chất lượng tốt 🡪 cần kiểm thử là đương nhiên

**Lỗi trong sản phẩm phần mềm**

* Lỗi trong phần mềm có thể dẫn đến:
  + Mất tiền để xử lý, khắc phục
  + Thời gian để xử lý, khắc phục
  + Mất sự tin tưởng từ phía khách hàng
  + Thương tổn về tài chính, con người.
* Các khái niệm về lỗi trong sản phẩm, dự án phần mềm:
  + Error: là chỉ hành động của con người gây ra một kết quả không đúng
  + Fault: là một bước sai, quy trình sai hay định nghĩa dữ liệu sai, fault là kết quả của error
  + Failure: là lỗi tìm được trong quá trình sản phẩm phần mềm đã được đưa vào sử dụng, đó chính là độ lệch (GAP) giữa sản phẩm và mong muốn của người sử dụng
* Lỗi xuất hiện khi nào:
  + Bất cứ khi nào tạo ra một sản phẩm thì đều có khả năng sản phẩm đó chứa lỗi
* Chi phí khắc phục lỗi thay đổi như thế nào?
  + Chi phí khắc phục, xử lý lỗi sẽ tăng dần theo thời gian
  + Những lỗi mà ở giai đoạn live use mới được phát hiện, nếu chiếu theo thuật ngữ về lỗi thì sẽ là failure, là những lỗi tốn nhiều chi phí nhất để sửa chữa và khắc phục

1 C

2 C

3 A

4 B

5 C

**--------------- --------------- --------------- --------------- --------------- ---------------**

**Bài 3: các chân lý trong kiểm thử phần mềm**

**Chân lý thứ nhất: testing shows presence of defects (kiểm thử sẽ chỉ ra các lỗi hiện diện hay việc kiểm thử sẽ tìm ra lỗi)**

* Testing sẽ chỉ ra có lỗi tồn tại trong hệ thống, chứ không chứng minh rằng sản phẩm phần mềm đó không có lỗi, hay chứng minh rằng sản phẩm phần mềm đó hoạt động đúng
* Testing làm giảm xác suất lỗi chưa tìm thấy, nhưng ngay cả khi không tìm thấy lỗi, nó cũng không chứng minh được rằng sản phẩm phần mềm đó hoạt động chính xác và không có lỗi

**Chân lý thứ hai: kiểm thử tất cả các trường hợp là không khả thi**

* Kiểm thử mọi thứ là không thể vì thời gian và chi phí sẽ tăng theo cấp số nhân
* Thay vì kiểm thử toàn bộ, việc phân tích rủi ro và dựa trên mức độ ưu tiên chúng ta có thể tập trung kiểm thử vào một số điểm cần thiết

**Chân lý thứ 3: kiểm thử sớm**

* Các hoạt động kiểm thử nên bắt đầu càng sớm càng tốt trong quy trình phát triển và nên tập trung vào các mục tiêu đã định ra từ trước
* Thực hiện kiểm thử sớm giúp phát hiện các lỗi sớm, chi phí để sửa chữa, khắc phục các lỗi đó sẽ rẻ hơn

**Chân lý thứ 4: Defect clustering (sự tập trung lỗi)**

* Trong quá trình thực hiện kiểm thử phần mềm, sẽ có lúc chúng ta thẫy lỗi tập trung ở một số ít các module trong sản phẩm, thông thường là những chức năng chính đòi hỏi xử lý phức tạp, hoặc chức năng do người có ít kinh nghiệm thực hiện

**Chân lý thứ 5: Pesticide Paradox (nghịch lý thuốc trừ sâu)**

* Nếu việc kiểm thử lặp đi lặp lại nhiều lần, thì cuối cùng sẽ có 1 số trường hợp kiểm thử không còn tìm thấy lỗi mới nào 🡺 nghịch lý thuốc trừ sâu. Để khắc phục điều này, cần xem xét sửa đổi các tình huống kiểm thử, các test cases cũng cần được “làm mới” để tìm ra các lỗi tiềm ẩn hơn nữa

**Chân lý thứ 6: Testing is context dependent (kiểm thử theo ngữ cảnh)**

**-** Với mỗi sản phẩm khác nhau, trong những lãnh vực khác nhau thì việc kiểm thử cũng sẽ khác nhau

**Chân lý thứ 7: sự sai lầm về việc không có lỗi**

* Việc tìm và sửa chữa lỗi sẽ là vô nghĩa nếu hệ thống xây dựng xong không thể dùng được và không đáp ứng được nhu cầu và mong đợi của người dùng
* Nghĩa là phần mềm, sau khi đã làm đầy đủ công đoạn để phát triển ra một sản phẩm phần mềm hoàn chỉnh nhưng sản phẩm đưa ra đến tay người dùng không đáp ứng được nhu cầu của người sử dụng thì phần mềm đó coi như thất bại, mặc dù các hoạt động đảm bảo chất lượng phần mềm, kiểm thử vẫn diễn ra đầy đủ

**1 D**

**2 D**

**3 C**

**4 C**

**5 B**

**--------------- --------------- --------------- --------------- --------------- --------------- --------------- --------------- --------------- --------------- --------------- ---------------**

**Phần 2: quy trình kiểm thử phần mềm căn bản**

**Bài 4: lập kế hoạch và kiểm soát hoạt động kiểm thử**

**Quy trình kiểm thử căn bản**

* Một quy trình kiểm thử căn bản được định nghĩa gồm 5 hoạt động chính:
  + Lên kế hoạch và kiểm soát hoạt động kiểm thử sản phẩm, hệ thống phần mềm – Test planning & control
  + Phân tích và thiết kế các tình huống kiểm thử - Test analyst & design
  + Viết các tình huống kiểm thử, chuẩn bị dữ liệu kiểm thử (implementation) và chạy chương trình để kiểm thử - test implementation & execution
  + Đánh giá tiêu chí hoàn thành và báo cáo kết quả kiểm thử - Evaluating exit & reporting
  + Tổng kết quá trình kiểm thử - Test closure activities

**Lên kế hoạch kiểm thử sản phẩm**