1. **Tại sao kiêm thử là cần thiết.**

* Kiểm thử phần mềm là cần thiết bởi vì mọi người đều mắc những sai lầm. Một vài lỗi thì không quan trọng, nhưng một vài lỗi thì thực sự là đắt giá và đáng nguy hiểm. Chúng ta cần kiểm tra mọi thứ và bất kì thứ gì chúng ta tạo ra bời vì mọi thứ đề có thể có những sai xót. “Humans make the mistakes all the time”
* Những nguyên nhân tại sao kiêm thử phần mềm là quan trọng.

1. Nó chỉ ra những khuyết điểm và sai sót đã được thực hiện trong các giai đoạn phát triển.
2. Nó là điều cần thiết vì nó đảm bảo độ tin cậy của khách hàng và sự hài lòng của họ trong ứng dụng.
3. Nó quan trọng để đảm bảo chất lượng của sản phẩm. Chất lượng sản phẩm giao cho khách hàng giúp cho việc đạt được sự tin cậy của họ.
4. Thử nghiệm là cần thiết để cung cấp các tính năng chính sác cho khách hàng.
5. Đó là nguyên tắc trong kinh doanh.

* Tại sao lại sảy ra lõi trong phần mềm.

1. Phần mềm được viết bởi con người.

Bạn có thể biết về một vài thứ, nhưng không phải là tất cả, bạn có kỹ năng nhưng kĩ năng của bạn chưa hoàn hảo và một điều quan trọng là mọi người đều mắc những sai lầm.

1. Áp lực phải hoàn thành phần mềm kịp deadline.

Bạn không có thời gian để kiểm tra toàn diện các lỗi, hệ thống của bạn chưa hoàn chỉnh nhưng deadline đã đến và bạn phải giao sản phẩm cho khách hàng.

* Giá của những lối phần mềm đôi khi nó rất ít hoặc không đáng giá nhưng đôi khi nó là một cái giá đắt sau đây và một vài vd:

1. 28/6/1962: Tàu thăm dò vũ trụ Mariner I có sứ mệnh bay đến sao Kim nhưng đã bị phá hủy 293 giây sau khi phóng. Nguyên nhân do một công thức được viết trên giấy bằng bút chì đã không được chuyển đổi sang mã điện toán, khiến hệ thống máy tính tính toán sai đường đi của tên lửa
2. 1988: Lỗi tràn bộ đệm trong trình Berkerley Unix. Morris, sâu Internet đầu tiên đã lây lan tới khoảng 2000-6000 máy tính của nhiều tổ chức ở mỹ trong vòng chưa đầy một ngày bằng cách tận dụng lỗi tràn bộ nhớ trung gian.
3. **Các nguyên tắc kiểm thử**

Nguyên tắc 1: Kiểm thử dưa ra lỗi.

Kiểm thử có thể cho thấy rằng phần mềm đang có lỗi, nhưng không thể chứng minh rằng phần mềm không có lỗi. Kiểm thử làm giảm xác suất lỗi chưa tìm thấy và vẫn còn trong phần mềm, thậm chí là không có lỗi nào, nó không phải là bằng chứng của sự chính xác.

Nguyên tắc 2: Kiểm thử đầy đủ là không thể

Kiểm thử mọi thứ là không thể thực hiện được. Thay vì kiểm thử toàn bộ, việc phân tích rủi ro và dựa trên mức độ ưu tiên chúng ta có thể tập chung việc kiểm thử vào một số điểm cần thiết.

Nguyên tắc 3: Kiểm thứ sớm

Để tìm được bug sớm, các hoạt động kiểm thử nên được bắt đầu càng sớm càng tốt trong quy trình phát triển phần mềm hoặc hệ thống, và nên tập trung vào các hoạt động đã định trước.

Nguyên tắc 4: Sự tập trung của lỗi

Nỗ lực kiểm thử nên tập trung một cách cân đối vào mật độ lỗi dự kiến và lỗi phá hiện ra sau đó trong các module. Một số ít các module thường chứa nhiều lỗi không phát hiện ra trong lúc kiểm thử trước khi phát hành, hoặc chịu trách nhiệm cho hầu hết các lỗi hoạt động của phần mềm.

Nguyên tắc 5: Nghich lý của thuốc trừ sâu

Nếu việc kiểm thử tương tự nhau được lặp lại nhiều lần thì cuối cùng sẽ có một số trường hợp kiểm thử sẽ không tìm thấy bất kì lỗi nào mới. Để khắc phục các trường hợp kiểm thử cần phải được xem xét và sửa đồi thường xuyên, và cần phải viết các test case mới và khác nhau để thực hiện nhiều phần khác nhau của phần mềm hoăc hệ thống để tìm ra lỗi tiềm ẩn nhiều hơn nữa.

Nguyên tắc này giống như việc trừ sâu trong nông nghiệp, nếu chúng ta cứ phun một loại thuốc với nồng độ giống nhau trong một khoảng thời gian dài thì có một số con sâu sẽ quen dần và cuối cùng việc phun thuốc giống như là tắm chúng vậy(nhờn thuốc) lúc đó chúng ta không thể diệt sạch chúng được. Do vậy để diệt sạch sâu một cách hiệu quả, người ta thường thay đổi các loại thuốc trừ sâu, mỗi loại chỉ dùng trong một khoảng thời gian ngắn.

Nguyên tắc 5: Kiểm thử theo cách ngữ cảnh độc lập

Nguyên tắc này là việc testing phụ thuộc vào ngữ cảnh, test trong nhiều ngữ cảnh khác nhau.

Để hiểu rõ hơn chúng ta xét ví dụ sau:

Vd: Một trương trình có tên “Calculator” có nhiều chức năng nhưng:

* Nếu test chương trình này cho mẫu giáo thì chỉ cần test cộng trừ là được.
* Nếu test chương trình này cho cấp 2 thì cộng trừ nhân chia.
* Nếu test chương trình này cho đại học thì tích phân, đạo hàm…

Nguyên tắc 7: Sự sai lầm về việc không có lỗi

Việ tìm và sửa chữa lỗi sẽ không giúp được gì nếu hệ thống được xây dựng xong nhưng không thẻ dùng được và không đáp ứng được nhu cầu của người dùng.

1. **Quy trinh kiểm tra cơ bản**.

Các bước trong một quy trình kiểm thử phần mềm cơ bản bao gồm:

* Lập kế hoạch kiểm tra
* Thiết kế test case
* Phát triển test script
* Thực hiện test
* Đánh giá kết quả test
* Lập kế hoạch : Nhằm chỉ định và mô tả các loại kiểm tra sẽ được triển khai và thực hiện. Kết quả của bước lập kế hoạch là bản tài liệu kế hoạch KTPM bao gồm nhiều chi tiết từ các loại kiểm tra, chiến lược kiểm tra, cho đến thời gian và phân định lực lượng kiểm tra viên.

1. Thiết kế test: Nhằm chỉ định các test case và các bước kiểm tra chi tiết cho mỗi phiên PM. Giai đoạn thiết kế test là hết sức quan trọng, hết sức quan trọng , nó đảm bảo tất cả các tình huống kiểm tra hết tất cả các yêu cầu

Các bước thiết kế test :

* + Xác định và mô tả test cây
  + Mô tả các bước chi tiết để kiếm tra
  + Xem xét và khảo sát độ bao phủ của việc kiểm tra
  + Xem xét test case và các bước kiểm tra

1. Phát triển test Script: Bước này thường không bắt buộc trong các loại và mức kiểm tra, chỉ yêu cầu trong những trường hợp đặc thù cần thiết kế, tạo ra các test script có khả năng chạy trên máy tính giúp tự động hoá việc thực thi các bước kiểm tra đã định nghĩa ở các bước thiết kế test
2. Thực hiện test

Mục đích thực hiện các bước kiểm tra đã thiết kế và ghi nhận kết quả

1. Đánh giá test

Mục đích: Đánh giá toàn bộ quá trình kiểm tra bao gồm xem xét và đánh giá kết quả kiểm tra lỗi, chỉ định các yêu cầu thay đổi và tính toán số liệu liên quan, đến quá trình kiểm tra.

1. **Psychology Of testing.**

* Độ tin cậy của phần mềm là độ đo về mức độ tốt của các dịch vụ mà hệ cung cấp cho máy tính. Cần chú ý là người dùng không xét rằng các dịch vụ là quan trọng như nhau.
* Có 4 thuộc tính quan trọng trong việc quyết định tính tin cậy của một phần mềm:
* Availability (Tính sẵn có): là thước đo về khả năng cung cấp các tính năng một cách liên tục cho người dùng đảm bảo không bị gián đoạn trong một khoảng xác định (có thể là thời gian hay số lượng yêu cầu)
* Reliability (Độ tin cậy): là thước đo về sự chính xác, hiệu quả trong việc cung cấp các tính năng người dùng mong muốn trong một khoảng thời gian sử dụng xác định.
* Safety (Sự an toàn): đảm bảo khi hoạt động hoặc khi bị sự cố phần mềm không gây ra thiệt hại cho người dùng.
* Security (Khả năng bảo mật): chống các can thiệp xấu từ bên ngoài nhằm phá hoại phần mềm hoặc đánh cắp thông tin.
* Vai trò của 1 tester trong việc kiêm thử
* Đưa ra thông tin chính xác về lỗi và những thay đổi
* Hiểu quan niệm từ các nhà phát triển về lĩnh vực phần mềm
* Đưa code test đến một tiêu chuẩn thống nhất
* Tìm ra lỗi
* Yêu cầu kĩ thuật và kế hoạch test
* Thực hiện nghiêm túc báo cáo về lỗi
* Đưa ra các dự đoán về mức độ lỗi trong tương lai
* Cải thiện trình độ test
* Trách nhiệm của testers
* Theo dõi kế hoạch test, bàn thảo…giống như hồ sơ.
* Báo cáo lỗi khách quan và chính xác
* Kiểm tra lỗi có đúng không trước khi báo cáo
* Đánh giá khách quan rủi ro
* Thận trọng với những gì bạn báo cáo
* Báo cáo sự thật