Lab1(<https://docs.google.com/document/d/1sj5gICuMV7kc1PXnZq2oLnAptry1lSFSrsvjB2Pcak8/edit?usp=sharing>)

**Tình huống1 : Nhiều người vẫn có quan điểm về nghề kiểm thử phần mềm, phổ biến nhất vẫn là suy nghĩ “Kiểm thử phần mềm, ai làm chẳng được”. Nhóm hãy thảo luận về quan điểm này.**

Bạn 1(Trâm): Theo ý kiến của tôi thì tôi đồng ý với quan điểm này vì trong thời điểm hiện tại thì tôi chỉ đang ở trong môi trường cao đẳng với những thành viên trong nhóm là những sinh viên với trình độ ngang nhau và dự án không đòi hỏi quá cao về vấn đề các chức năng của phần mềm. Vì vậy việc thực kiểm thử phần mềm trong môi trường cao đẳng thì “Ai làm cũng được”.

Bạn 2(Kha): Tôi vừa đồng ý và cũng không đồng ý quan điểm trên bởi vì nó có 2 luồng khác nhau:

Đồng ý là vì: Test thì chúng ta chỉ cần kiểm tra các chức năng của chúng và ghi lại và chụp lại các lỗi sai của phần mềm,

Không đồng ý là vì: Là người kiểm thử **phần mềm** , bạn tham gia vào giai đoạn đảm bảo chất lượng của quá trình phát triển và triển khai **phần mềm** . Bạn sẽ tiến hành kiểm tra tự động và thủ công để đảm bảo **phần mềm** do nhà phát triển tạo ra phù hợp với mục đích sử dụng. ... Đánh giá mã là một phần trong vai trò của người kiểm thử **phần mềm** .

Muốn làm test thì phải có trình độ chuyên môn nhất định, nếu như không có trình độ chuyên môn thì sẽ không biết nó sai ở điểm nào và cần phải khắc phục ở điểm

Với tét một sản phẩm trong thời gian hai giờ, tét có trình độ chuyên môn nhất định thì có thể test đến 50 testcases nhưng Tester chỉ có kỹ năng kiểm thử thì chỉ kiểm thử vài ba cases,Tester cần phải hiểu sâu rộng về lĩnh vực hoạt động của sản phẩm, về người dùng cuối, cũng như kiến thức phát triển phần mềm. Và cả hai trường hợp điều này có thể vẫn còn sót bug. Cho thấy rằng việc test không dễ dàng như chúng ta nghĩ muốn làm tét thì phải có trình độ chuyên môn nhất định

Bạn 3(Hà):Theo ý kiến của tôi thì quan điểm này không đúng vì để có thể test 1 phần mềm mà đáp ứng việc nó hạn chế có lỗi hay khuyết điểm thì yêu cầu người kiểm thử phải có sự hiểu biết hay nắm được kiến thức kiểm. Đồng thời nếu một người nào đó không hiểu về kiểm thử thì họ sẽ không thể chắn chắc rằng quá trình hay việc mình kiểm là đang đúng . Vì vậy theo tôi việc kiểm thử phần mềm yêu cầu phải có trình độ và kiến thức.

**Tình huống 2: “Nghề kiểm thử không đòi hỏi nhiều khả năng phân tích và sáng tạo”. Nhóm hãy thảo luận về câu trên ?**

Bạn Hà : Theo tôi nghĩ thì nghề kiểm thử không đòi hỏi khả năng phân tích và sáng tạo là đúng vì ở đây khi ta tiến hành kiểm thử một phần mềm thì đã có sẵn những test case và test unit đã được viết sẵn các tester chỉ cần làm theo và xác nhận xem nó có như những gì đã nêu ra là được

Bạn Kha:

Bạn Trâm : Tôi không đồng ý với ý kiến trên vì nghề kiểm thử cần đòi hỏi

**Tình huống 3: Các nhóm hãy thảo luận và đưa ra những tố chất để trở thành một người Kiểm thử và dựa vào các tố chất này so sánh với bản thân xem có phù hợp với ngành kiểm thử hay không ?**

-Những tố chất của 1 tester

* Tò mò
* Chú ý tiểu tiết
* Trí tưởng tượng
* Tư duy logic
* Khả năng tập trung và phân tích
* Kỉ luật
* Giao tiếp có tính xây dựng
* [Bạn giỏi báo cáo](https://www.softwaretestinghelp.com/10-qualities-that-can-make-you-a-good-tester/#6_You_Are_Good_At_Reporting)
* [Bạn có thể linh hoạt hỗ trợ bất cứ khi nào cần thiết](https://www.softwaretestinghelp.com/10-qualities-that-can-make-you-a-good-tester/#7_You_Are_Flexible_To_Support_Whenever_Its_Required)
* [Thái độ tích cực](https://www.softwaretestinghelp.com/10-qualities-that-can-make-you-a-good-tester/#1_Positive_Attitude)
* [Giao tiếp tốt](https://www.softwaretestinghelp.com/10-qualities-that-can-make-you-a-good-tester/#2_Good_Communication)
* [Khả năng đa tác vụ](https://www.softwaretestinghelp.com/10-qualities-that-can-make-you-a-good-tester/#3_Multi-Tasking_Abilities)
* [Đam mê thử nghiệm](https://www.softwaretestinghelp.com/10-qualities-that-can-make-you-a-good-tester/#5_Passion_For_Testing)
* [Khả năng phân tích](https://www.softwaretestinghelp.com/10-qualities-that-can-make-you-a-good-tester/#8_Analytical_Abilities)
* [Thực hành và có sự đồng cảm](https://www.softwaretestinghelp.com/10-qualities-that-can-make-you-a-good-tester/#10_Practice_Empathy)

Bạn Hà: Sau khi đã tìm hiểu và biết được những tố chất cần có của một tester em cảm thấy rằng bản thân của mình không phù hợp với ngành kiểm thử vì em không thấy ở mình có một tố chất nào mà một tester cần có.

**Bạn Trâm : Em cảm thấy em có 1 vài tố chất đã kể trên như tò mò, chú ý tiểu tiết, khả năng tập trung và phân tích nhưng lại thiếu yếu tố quan trọng là đam mê với việc kiểm thử vì theo ý kiến của em thì việc kiểm thử là công việc phải thử nghiệm nhiều lần dễ dẫn đến nhàm chán.**

**Lab1:**

**Phần 1:**

**Câu 1: Sau khi đã học và hiểu vì sao phải kiểm thử phần mềm thì trước đây trong quá trình**

**làm việc với những dự án hay bài tập lớn do nhà trường yêu cầu bạn đã không áp dụng khái niệm, nguyên lý, quy trình gì khi thực hiện phát triển phần mềm ?**

Sau khi đã học và hiểu vì sao phải kiểm thử phần mềm. Trước đây trong quá trình làm việc với những dự án như Dự Án 1 nhóm em đã không áp dụng các quy trình:

o Nhiệm vụ chính Lập kế hoạch kiểm thử

o Kiểm soát và giám sát kiểm thử(Test control)

o Viết các trường hợp test (Test Case)

o Thiết kế kiểm thử(Test Design)

o Triển khai kiểm thử(Test Implementation)

o Tiêu chí đánh giá kết thúc và báo cáo

Việc kiểm thử của nhóm chỉ dựa trên các lỗi có thể thấy từ các chức năng của phần mềm, và các lỗi mà nghĩ có thể xảy ra lúc đó.

**Câu 2: Công ty XY muốn làm ra phần mềm quản lý nhân sự có tất cả đầy đủ tính năng và**

**đảm bảo 95% các tính năng đều vận hành tốt không xảy ra lỗi. Bạn hãy đánh giá**

**xem khi nào là Xác nhận(Validation) và Xác minh(Verification)**

**Trả lời:**

Xác nhận(Validation): phần mềm quản lý nhân sự có tất cả đầy đủ tính năng.

Xác minh(Verification): đảm bảo 95% các tính năng đều vận hành tốt không xảy ra lỗi.

**Phần 2:**

**Câu 4: Tuần 1: Bạn tìm thấy khiếm khuyết với mức độ nghiêm trọng 1 vào ngày 1 và việc kiểm thử hoàn chỉnh bị chặn trong 3 ngày. Do đó bạn không thể thực hiện bất kỳ kịch bản cho đến khi khuyếm khuyết mức độ nghiêm trọng 1 đó được giải quyết. Sau khi mất 3 ngày giải quyết nó, bạn tiếp tục với thực hiện việc kiểm thử của mình. Vào cuối tuần, bạn hoàn thành 20 kịch bản với nhiều khiếm khuyết hơn. Tuần này vẫn giống như kịch bản 1.**

**Tuần 2: Bạn tiếp tục có một vài khiếm khuyết mức độ nghiêm trọng 2 và mức độ nghiêm trọng 3 trong tuần thứ hai, do đó bạn không thể bao phủ hết kịch bản để đã bị tồn đọng từ tuần 1.**

**–Với trường hợp trên hãy liệt kê các tiêu chí kết thúc chưa hoàn thành**

**Trả lời :**

* Quá nhiều lỗi trong và nhiều lỗi nghiêm trọng
* Không đủ thời gian cho việc hoàn thành kiểm các kịch
* bản
* Chưa có môi trường phù hợp cho việc kiểm thử
* Phần mềm chưa sẵn sàng cho việc kiểm

**Câu 5:Một nhóm kiểm thử luôn tìm ra số lượng lớn lỗi trong suốt quá trình phát triển, kể cả kiểm thử hệ thống. Mặc dù người quản lý kiểm thử hiểu rằng việc phát hiện lỗi này khá tốt trong phạm vi ngân sách của nhóm kiểm thử cô phụ trách, song lãnh đạo cấp cao vẫn chưa hài lòng, phàn nàn rằng nhóm kiểm thử đã bỏ sót vài lỗi mà người dùng phát hiện ra sau khi chuyển giao sản phẩm. Giả dụ người dùng nhìn chung là hài lòng với hệ thống và hỏng hóc chỉ có ít ảnh hưởng, thì nguyên lý kiểm thử nào có thể giúp người quản lý kiểm thử giải thích với lãnh đạo về lý do bỏ sót một số lỗi?**

**Trả lời**

### **Kiểm thử toàn bộ là không khả thi**:

### Đúng vậy, rất khó để kiểm tra toàn bộ các module cũng như các tính năng, kết hợp với đầu vào và đầu ra trong suốt quá trình kiểm tra. Các sản phẩm phần mềm hiện nay cực kỳ đa dạng và phức tạp, được phát triển trên nhiều nền tảng, thêm vào đó, ngày càng có nhiều công nghệ mới, khả năng kết nối dữ liệu lớn… khiến việc kiểm thử toàn bộ gần như là không thể. Thay vì cố gắng kiểm thử toàn bộ, hãy xác định mức độ quan trọng và độ ưu tiên của các module để kiểm thử những phần cần thiết hoặc gặp nhiều nguy cơ hơn.

### **Lỗi thường phân bổ tập trung:**

### Nguyên lý về phân bố lỗi chỉ ra rằng, chỉ một số ít module chứa phần lớn số lỗi phát hiện được. Những module này thường là những thành phần, chức năng chính của hệ thống. Điều này cũng đúng theo nguyên lý Pareto: 80 – 20: 80% số lỗi tìm thấy ở chỉ 20% module. Bằng kinh nghiệm, các QA/ Tester có thể xác định được những module có tính rủi ro và nhiều lỗi như vậy, giúp tập trung tìm kiếm lỗi nhanh và hiệu quả hơn. Tuy nhiên, cách tiếp cận này cũng ẩn chứa vấn đề: nếu thực hiện kiểm thử tương tự lặp đi lặp lại, dễ thấy rằng những test case cũ sẽ khó tìm thêm được bug mới.

### **Quan niệm sai lầm về việc “hết lỗi”:**

### Một phần mềm sạch bug 99% vẫn có thể không sử dụng được. Đây là trường hợp phần mềm được kiểm thử bằng một requirement sai. Kiểm thử không chỉ để tìm ra lỗi, mà còn để kiểm tra xem phần mềm có đáp ứng được đúng nhu cầu hay không. Chính vì vậy, việc Không còn lỗi hay Hết lỗi là quan niệm sai lầm.

### 