**MODULE 1. TÌM LỖI CHƯƠNG TRÌNH**

## Bài 1.

1. **Đưa ra một số lý do giải thích tại sao đặc tả là nơi chứa nhiều bug nhất.**

* Do người làm phần mềm không nắm rõ hệ thống mình đang phát triển
* Do người đặt hàng nhiều khi là các nhà quản lý, không phải là người dùng trực tiếp nên phát biểu yêu cầu thường không chính xác , không rõ rang
* Do sự hiểu lầm trong giao tiếp hoặc không giao tiếp : Các yêu cầu không rõ ràng và giải thích sai các yêu cầu là 2 yếu tố chính gây ra lỗi.

1. **Giải thích tại sao chi phí sửa bug sau khi sản phẩm release lại cao hơn khi mới bắt đầu.**

* Lỗi được phát hiện càng muộn thì chi phí cho việc sửa lỗi càng lớn, bởi vì có những lỗi sẽ phải thực hiện lại từ khâu Thiết kế, rồi coding lại và mới thực hiện test được. Nên lỗi được phát hiện càng sớm, càng ở những giai đoạn đầu dự án, thậm chí ngay từ giai đoạn làm Yêu cầu/ Nghiệp vụ giúp cho các giai đoạn sau thực hiện được chính xác, giảm được số lượng lỗi và sản phẩm hoàn thành đúng tiến độ theo kế hoạch.
* Bug phát sinh ở giai đoạn release là nghiêm trọng và tốn kém nhất. Không chỉ bị ảnh hưởng về mặt uy tín chất lượng sản phẩm, mà còn dẫn đến việc phải coding và testing lại, phát sinh chi phí về nhân lực dự án, chậm trễ tiến độ.
* Bug được phát hiện càng muộn thì chi phí sửa càng lớn. Đôi khi không chỉ tỷ lệ với thời gian mà có thể là tỷ lệ bình phương thời gian.

1. **Nêu và giải thích 7 nguyên tắc của kiểm thử.**

* **Principle** **1** – Testing cho\_thấy sự hiện diện của defects, làm người thấy được lỗi : kiểm thử để cho thấy sự hiện diện của lỗi, chứ kiểm thứ không thể chứng minh rằng phần mềm không có lỗi. Lỗi luôn hiện diện trong phần mềm, chỉ là chưa đủ điều kiện để xuất hiện. Do đó khi kiểm thử tìm được càng nhiều lỗi càng tốt.
* **Principle 2** – Kiểm thử toàn diện là không thể (Exhaustive testing = complete testing) : kiểm thử toàn diện là kiểm thử tất cả mọi thứ(tất cả tổ hợp của điều kiện input đầu vào) điều này là không khả thi trong hầu hết các trường hợp vậy nên điều chúng ta cần làm là tính toán mức độ ưu tiên để có được số lượng tối ưu các kiểm thử
* **Principle 3** – Testing sớm : nên thực hiện testing ở tất cả giai đoạn của quá trình phát triển phần mềm vì lỗi phát hiện càng muộn thì chi phí sửa chữa càng lớn.
* **Principle 4** – Phân cụm lỗi : theo nguyên tắc PARETO (Luật 80-20) – xấp xỉ 80% vấn đề được phát hiện trong khoảng 20 % module.
* **Principle 5** – Nghịch lý thuốc trừ sâu : Để diệt sâu một cách hiệu quả, người ta thường thay đổi loại thuốc trừ sâu, mỗi loại chỉ dùng trong một khoảng thời gian ngắn. Tương tự tester cần luôn luôn làm mới bộ dữ liệu test
* **Principle 6** – Testing phụ thuộc ngữ cảnh : tùy thuộc vào yêu cầu, chức năng và mục đích sử dụng của hệ thống phần mềm sẽ có các phương pháp test khách nhau
* **Principle 7** – Sự sai lầm về việc không có lỗi : một hệ thống phần mềm không có lỗi không đồng nghĩa với việc đáp ứng được nhu cầu và sự mong đợi của người dùng (khách hàng) – gây ra sự thất bại cho dự án phần mềm

## Bài 2.

**Hãy tìm các defect/bug (càng nhiều càng tốt) trong hình dưới đây. Các lỗi có thể là giao diện không nhất quán, lỗi chính tả, trùng lắp,…**



**8** Thông báo : “Số đo nhiệt độ tùy ý(số nguyên) trong khoảng -60°C đến +60°C”

**7**Thông báo : “Số đo nhiệt độ tùy ý(số nguyên) trong khoảng -60°C đến +60°C”

**6** Thông báo : “Số đo nhiệt độ tùy ý(số nguyên) trong khoảng -60°C đến +60°C”

**5** Thông báo : “Số đo nhiệt độ tùy ý(số nguyên) trong khoảng -60°C đến +60°C”

**4** Thông báo : “Số đo nhiệt độ tùy ý(số nguyên) trong khoảng -60°C đến +60°C”

**3** Thông báo : “Số đo nhiệt độ tùy ý(số nguyên) trong khoảng -60°C đến +60°C”

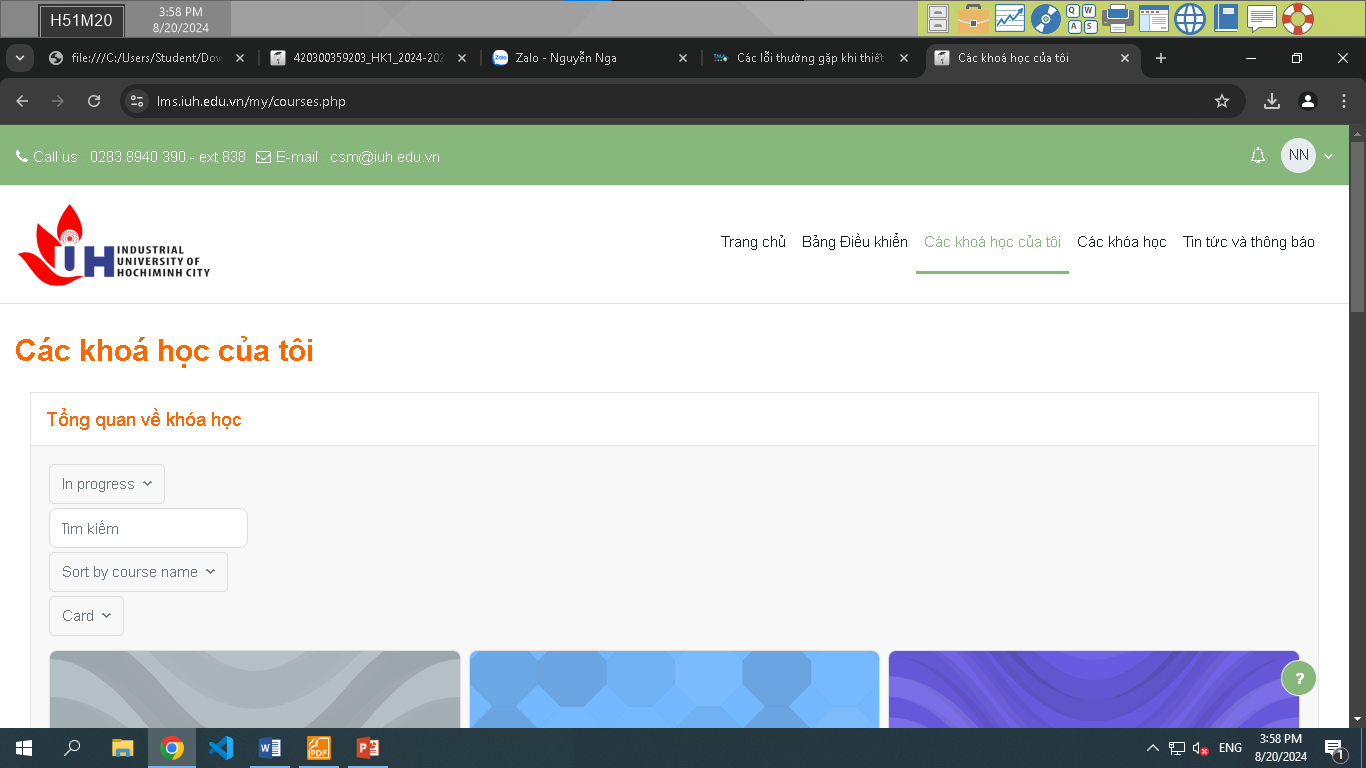
**2** Thông báo : “Số đo nhiệt độ tùy ý(số nguyên) trong khoảng -60°C đến +60°C”

**1** Thông báo : “Số đo nhiệt độ tùy ý(số nguyên) trong khoảng -60°C đến +60°C”

1. Lỗi chính tả “how many defects can you find here” : viết hoa chữ cái đầu câu how => How
2. Lỗi chính tả “Untitled)” => “(Untitled)”
3. Lỗi chính tả “Gooogle Search” => “Google Search”
4. Lỗi trùng lặp “Done Done”
5. Lỗi không nhất quán “more” => “More”
6. Lỗi không nhất quán “Search settings’ => “Search Settings” (nhất quán với “Sign In”)
7. Lỗi bố cục giao diện không hợp lý : vị trí link “News” nằm phía trên so với các link khác
8. Lỗi Button không phù hợp với bối cảnh “I’m Feeling Unlucky”

## Bài 3.

**Hãy chỉ ra ít nhất một lỗi bất kỳ trong ứng dụng hoặc website bạn đã từng sử dụng.**

****

**1** Thông báo : “Số đo nhiệt độ tùy ý(số nguyên) trong khoảng -60°C đến +60°C”

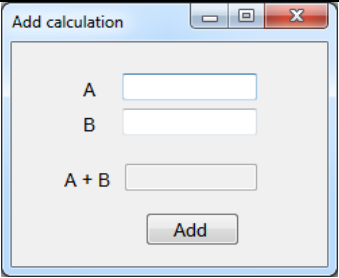
1. Lỗi không nhất quán ngôn ngữ trong cùng một tác vụ lọc/tìm kiếm:

“Tìm kiếm” – Tiếng việt

“In progress”, “Sort by course name”, “Card” – Tiếng anh

## Bài 4.

**Tìm các trường hợp kiểm thử (test case) cho chương trình cộng 2 số nguyên có giao diện sau:**



|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Dữ liệu nhập** | | **Kết quả mong muốn** |
| **A** | **B** | **A + B** |
| 1 | 1 | 2 | 3 |
| 2 | @ | 1 | Thông báo : “A không hợp lệ, vui lòng nhập lại.” |
| 3 | 1 | ! | Thông báo : “B không hợp lệ, vui lòng nhập lại.” |
| 4 | Rỗng | 1 | Thông báo : “Vui lòng nhập số nguyên A” |
| 5 | 1 | Rỗng | Thông báo : “Vui lòng nhập số nguyên B” |
| 6 | 1.5 | 2.5 | Thông báo : “Vui lòng nhập số nguyên ” |
| 7 | -5 | -2 | -7 |
| 8 | Rỗng | Rỗng | Rỗng |

## Bài 5.

**Cho chương trình đọc một loạt các số đo nhiệt độ tuỳ ý (số nguyên) trong khoảng -60°C đến +60°C và in ra giá trị trung bình của các số này. Tìm các trường hợp cần kiểm thử cho chương trình này, lập theo bảng sau:**

|  |  |  |
| --- | --- | --- |
| **STT** | **Dữ liệu nhập** | **Kết quả mong muốn** |
| 1 | [20, 30, 10, 60] | 30 |
| 2 | [-60, 50, 30, 20] | 10 |
| 3 | [-70, 10, 30, 50, 65] | Thông báo : “Số đo nhiệt độ tùy ý(số nguyên) trong khoảng -60°C đến +60°C” |
| 4 | [A, B, C, D, 30] | Thông báo : “Số đo nhiệt độ tùy ý(số nguyên) trong khoảng -60°C đến +60°C” |
| 5 | [] | [] |