**1. Khi nào thì có thể dừng việc kiểm thử?**

Đã thực hiện hết các test-case được đề ra. Hoàn thiện các chức năng và toàn bộ code đã đảm bảo được các yêu cầu đề ra. Tỷ lệ bug phải trong giới hạn mong muốn. Không có những bug nghiêm trọng

**2. Việc kiểm thử có thể bắt đầu vào thời điểm nào?**

Việc kiểm thử có thể bắt đầu ngay từ giai đoạn phân tích yêu cầu và tiếp tục xuyên suốt các giai đoạn phát triển phần mềm, tùy vào loại kiểm thử cụ thể.

**3. Hoạt động Verification là gì?**

Verification là một quá trình đánh giá các sản phẩm làm việc trung gian của một vòng đời phát triển phần mềm để kiểm tra xem liệu rằng chúng ta có đi đúng hướng để tạo ra sản phẩm cuối cùng.

**4. Tester cần có các tố chất nào?**

* Đủ chuyên môn và **có** kiến thức về lập trình. ...
* Tỉ mỉ, sáng tạo. ...
* Biết cách đặt câu hỏi. ...
* Luôn học hỏi và tìm tòi. ...
* Khả năng tư duy phân tích. ...
* Kỹ năng giao tiếp. ...

**5. Tại sao ngoài việc áp dụng các kỹ thuật kiểm thử thì vẫn cần kỹ thuật đoán lỗi trong kiểm thử phần mềm?**

Kỹ thuật đoán lỗi trong kiểm thử phần mềm giúp dự đoán các lỗi tiềm ẩn dựa trên kinh nghiệm và hiểu biết về hệ thống, từ đó tập trung kiểm tra các khu vực dễ xảy ra lỗi, tiết kiệm thời gian và tăng hiệu quả kiểm thử mà các kỹ thuật kiểm thử thông thường có thể bỏ sót.

**7.Có thể kiểm thử hoàn toàn một sản phẩm phần mềm không? Giải thích tại sao?**

- Không. Vì sự phức tạp của phần mềm, Không thể kiểm tra tất cả các kết hợp dữ liệu,

Hạn chế về thời gian và nguồn lực, Có thể bỏ sót các tình huống bất ngờ

**7. So sánh việc kiểm thử trong các mô hình thác nước và mô hình chữ V?**

*Điểm giống nhau:*

* Phụ thuộc vào giai đoạn phát triển phần mềm
* Quy trình tuần tự

*Điểm khác nhau:*

| **Tiêu chí** | **Mô hình thác nước** | **Mô hình chữ V** |
| --- | --- | --- |
| **Thời điểm kiểm thử** | Kiểm thử chủ yếu bắt đầu sau khi phát triển xong sản phẩm. | Kiểm thử được lên kế hoạch song song với từng giai đoạn phát triển. |
| **Tích hợp kiểm thử** | Ít tập trung vào tích hợp kiểm thử với phát triển. | Tích hợp kiểm thử ngay từ đầu, gắn chặt giữa yêu cầu và kiểm thử. |
| **Hiệu quả phát hiện lỗi** | Dễ bỏ sót lỗi vì kiểm thử diễn ra muộn. | Giảm thiểu lỗi vì kiểm thử được xem xét ở từng giai đoạn thiết kế. |
| **Phản hồi và chỉnh sửa** | Khó sửa đổi khi phát hiện lỗi muộn. | Dễ dàng điều chỉnh hơn nhờ kiểm thử liên tục trong quy trình. |

**8. Một input nhận giá trị năm sinh trong khoảng [2000, 2010]. Các giá trị biên để kiểm thử cho input field này là?**

1999, 2000, 2001, 2009, 2010, 2011.

**9. Kể tên các kỹ thuật dùng trong Black box testing?**

* Phân vùng tương đương
* Phân tích giá trị biên
* Bảng quyết định
* Kỹ thuật chuyển đổi trạng thái

**10. Test field lấy năm sinh từ [1994- 2006]. Viết 3 test case cho 3 lớp tương đương?**

- Hợp lệ là trong khoảng 1994-2006

- Lớp hợp lệ:

* + Input: 2000
  + KQMD: Hệ thống chấp nhận.
* Lớp không hợp lệ (dưới biên):
  + Input: 1993
  + KQMD: Hệ thống báo lỗi.
* Lớp không hợp lệ (trên biên):
  + Input: 2007
  + KQMD: Hệ thống báo lỗi.

**11. Quy định Input của một phần mềm tuyển dụng như sau: Tuổi là số nguyên trong khoảng [18, 30] … hãy cho 3 test case phù hợp để kiểm thử theo kỹ thuật Equivalence partitioning?**

- Tương tự câu 10 . Lớp hợp lệ là từ 18-30, Test case 1: input = 19(true), Test case 2: input = 7(true), Test case 3: input = 32(false)

**12. Một ô nhập liệu Điểm có yêu cầu nhập điểm có phạm vi [0,10] cho phép nhập số lẻ làm tròn 1 số thập phân (ví dụ: 7.5), cho 3 test case phù hợp để kiểm tra ô nhập liệu trên?**

TC1 :Hợp lệ: Điểm = 7.5 (thuộc phạm vi [0, 10] và định dạng đúng).

TC2 :Không hợp lệ - nhỏ hơn giới hạn: Điểm = -1 (nhỏ hơn 0).

TC3 :Không hợp lệ - định dạng sai: Điểm = 5.55 (nhiều hơn 1 chữ số thập phân).

**13. Một ô input trong chương trình chỉ cho phép nhập 5 ký tự gồm cả chữ hoa, thường và ký tự số, hãy cho ít nhất 3 test case để kiểm tra ô input trên?**

Hợp lệ: Abc12 (đúng 5 ký tự, bao gồm chữ hoa, chữ thường, và số).

Không hợp lệ - ít hơn 5 ký tự: Ab1 (3 ký tự).

Không hợp lệ - nhiều hơn 5 ký tự: Abc123 (6 ký tự).

**14. Cho đoạn code sau đây, hãy thiết kế 2 test case để thỏa tiêu chí bao phủ câu lệnh?**

**int liability (int age, char sex, boolean married) { Premium=500;**

**if ((age<25) && (sex==male) && (married==false)) {**

**Premium += 1500;…}**

**}**

*Giải:*

- Test case 1 (điều kiện đúng):

* Input: age = 20, sex = male, married = false.
* Expected Output: Premium = 2000 (500 + 1500).

- Test case 2 (điều kiện sai):

* Input: age = 30, sex = female, married = true.
* Expected Output: Premium = 500.

**15. Compatibility testing là gì?**

- là quá trình kiểm tra tính tương thích của 1 ứng dụng hoặc hệ thống với 1 số yêu cầu và môi trường cụ thể

**16. Sự cần thiết của Configuration testing?**

-Là cần thiết để kiểm tra phần mềm hoạt động đúng trên các cấu hình phần cứng, phần mềm và môi trường khác nhau

-Giúp đảm bảo phần mềm có thể hoạt động ổn định và hiệu suất tốt trên nhiều hệ thống khác nhau, tránh các lỗi không mong muốn do sự thay đổi cấu hình

**17. Khi Configuration testing mà ta không đủ các loại phần cứng cần thiết để kiểm tra thì làm sao?**

- Sử dụng phần mềm giả lập

- Máy ảo

- Thuê dịch vụ đám mây

- Mượn hoặc chia sẻ phần cứng

**18. Ưu điểm chính của phương pháp Integration testing từ trên xuống dưới (Top-down) là gì?**

Thu gọn phạm vi kiểm tra, dễ dàng tìm bug hơn. Các module quan trọng được ưu tiên trước, lỗi quan trọng cũng có thể được tìm thấy sớm hơn.

**19. Định nghĩa về Test case là? Những nội dung cần có trong mỗi test case là gì?**

*Test case* là một bộ các điều kiện, đầu vào, và các bước kiểm thử được xác định trước để kiểm tra một chức năng hoặc tính năng cụ thể của phần mềm

*Những nội dung cần có trong mỗi Test Case:*

Test Case ID,Tên Test Case,Mô tả ,Điều kiện tiên quyết ,Dữ liệu đầu vào ,Các bước kiểm thử ,Kết quả mong đợi,Kết quả thực tế,Trạng thái ,Ghi chú

**20. Như thế nào là một Test case được đánh giá tốt?**

Rõ ràng và chi tiết,Độc lập,Mục tiêu rõ ràng,Dễ bảo trì,Kết quả dễ so sánh,Tái sử dụng,Bao quát các tình huống hợp lệ và bất hợp lệ