**Nhóm 3**

18110285 Nguyễn Thị Minh Hoàng

18110371 Trần Ngọc Minh Thiện

18110377 Nguyễn Huỳnh Minh Tiến

18110402 Lê Thị Ngọc Yến

**Bài tập**

***Kiểm thử phần mềm***

**------o0o------**

**Bài tập 01: Tổng quan về kiểm thử phần mềm**

1. **Tại sao chúng ta phải thực hiện kiểm thử phần mềm? Những vấn đề cơ bản mà nhiều phần mềm thường gặp phải là gì?**

* Mục đích của kiểm thử phần mềm là chỉ ra rằng phần mềm thực hiện đúng các chức năng mong muốn.
* Một số vấn đề thường gặp trong phát triển phần mềm:
  + Tính toán không đúng, hiệu chỉnh sai dữ liệu.
  + Trộn dữ liệu không đúng.
  + Tìm kiếm dữ liệu sai yêu cầu.
  + Xử lý sai mối quan hệ giữa các dữ liệu.
  + Coding/hiện thực sai các qui luật nghiệp vụ.
  + Hiệu suất của phần mềm còn thấp.
  + Kết quả hoặc hiệu suất phần mềm không tin cậy.
  + Hỗ trợ chưa đủ các nhu cầu nghiệp vụ.

1. **Mục tiêu của việc kiểm thử là gì? Công việc chính của nhân viên kiểm thử phần mềm (Software Tester) là gì?**

* Các mục tiêu chính của kiểm thử phần mềm :
  + Phát hiện càng nhiều lỗi càng tốt trong thời gian kiểm thử xác định trước.
  + Chứng minh rằng sản phẩm phần mềm phù hợp với các đặc tả yêu cầu của nó.
  + Xác thực chất lượng kiểm thử phần mềm đã dùng chi phí và nỗ lực tối thiểu.
  + Tạo các testcase chất lượng cao, thực hiện kiểm thử hiệu quả và tạo ra các báo cáo vấn đề đúng và hữu dụng.
* Công việc chính của tester là đảm bảo chất lượng của phần mềm, kiểm tra để phát hiện các lỗi đang tồn tại trước khi giao sản phẩn cho khách hàng, Tester sẽ làm việc cùng đội phát triển phần mềm để sản phẩm tới tay khách hàng hoàn thiện nhất.

1. **Phân biệt 2 thuật ngữ Verification và Validation trong kiểm thử phần mềm.**

Kiểm thử phần mềm là 1 thành phần trong lĩnh vực rộng hơn, đó là Verification & Validation (V &V), ta tạm dịch là Thanh kiểm tra và Kiểm định phần mềm.

* Thanh kiểm tra phần mềm là qui trình xác định xem sản phẩm của 1 công đoạn trong qui trình phát triền phần mềm có thoả mãn các yêu cầu đặt ra trong công đoạn trước không .Thanh kiểm tra phần mềm thường là hoạt động kỹ thuật vì nó dùng các kiến thức về các artifacts, các yêu cầu, các đặc tả rời rạc của phần mềm. Các hoạt ₫ộng Thanh kiểm tra phần mềm bao gồm kiểm thử (testing) và xem lại (reviews).
* Kiểm định phần mềm là qui trình đánh giá phần mềm ở cuối chu kỳ phát triển để đảm bảo sự bằng lòng sử dụng của khách hàng. Các hoạt động kiểm định được dùng để đánh giá xem các tính chất được hiện thực trong phần mềm có thỏa mãn các yêu cầu khách hàng và có thể theo dõi với các yêu cầu khách hàng không.

1. **Trình bày các công đoạn chính trong 1 quy trình phát triển phần mềm nói chung.**

* Đặc tả phần mềm: Định nghĩa các chức năng, điều kiện hoạt động của phần mềm.
* Phát triển phần mềm: Quá trình xây dựng các đặc tả.
* Đánh giá phần mềm: Phầm mềm phải được đánh giá để chắc chắn rằng ít nhất có thể thực hiện những gì mà tài liệu đặc tả yêu cầu.
* Tiến hóa phần mềm: Đây là quá trình hoàn thiện các chức năng cũng như giao diện để ngày càng hoàn thiện phần mềm cũng như các yêu cầu đưa ra từ phía khách hàng.

1. **Trình bày cơ bản về quy trình thác nước (Waterfall) và quy trình RUP.**

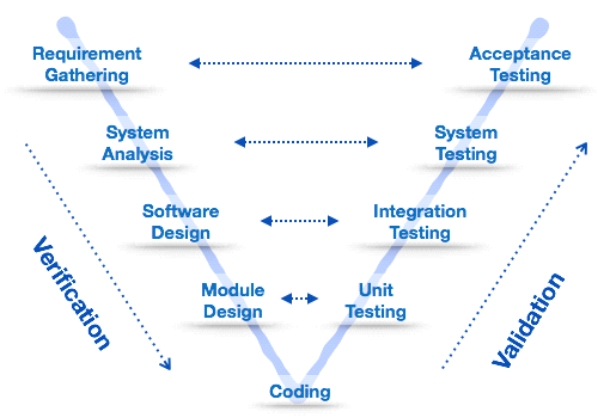
* **Quy trình thác nước (Waterfall)**
  + 1. Phân tích yêu cầu (Requirement Analysis): nắm được tất cả các yêu cầu, thảo luận và hướng dẫn để hiểu các yêu cầu. Thực hiện thử nghiệm các yêu cầu để đảm bảo rằng các yêu cầu có thể kiểm chứng được hay không.
  + 2. Thiết kế hệ thống (System Design): theo yêu cầu để tạo ra thiết kế. Thảo luận về yêu cầu phần cứng /phần mềm. Hoàn thành tài liệu thiết kế
  + 3. Thực hiện (Implementation): theo thiết kế tạo ra các chương trình/code. Tích hợp code cho giai đoạn tiếp theo. Unit testing
  + 4. Thử nghiệm hệ thống (System Testing): tích hợp unit tested code và kiểm tra nó để đảm bảo nó hoạt động như mong đợi. Thực hiện tất cả các hoạt động thử nghiệm (Functional and non functional) để đảm bảo hệ thống đáp ứng các yêu cầu. Trong trường hợp bất thường thì lập tức báo cáo. Theo dõi tiến độ về kiểm tra thông qua các công cụ như số liệu truy xuất nguồn gốc, ALM. Báo cáo hoạt động thử nghiệm
  + 5. Triển khai Hệ thống (System Deployment): chắc chắn rằng môi trường đang hoạt động và không có lỗi mở server. Đảm bảo rằng các tiêu chí test được đáp ứng. Triển khai ứng dụng trong môi trường tương ứng. Thực hiện kiểm tra về môi trường sau khi ứng dụng được triển khai để đảm bảo ứng dụng không gặp vấn đề
  + 6. Bảo trì hệ thống (System Maintenance): Chắc chắn rằng ứng dụng đang chạy và chạy trong môi trường tương ứng. Trong trường hợp người dùng gặp lỗi, chắc chắn giải quyết và khắc phục các vấn đề. Trong trường hợp bất kỳ vấn đề fixed; code cập nhật được triển khai trong môi trường. Ứng dụng luôn được tăng cường để kết hợp nhiều tính năng, cập nhật môi trường với các tính năng mới nhất
* **Quy trình RUP (Rational Unified Process)**

Bao gồm 4 pha: Inception(khởi đầu), Elaboration(Xây dựng phát thảo), Construction(Xây dựng), Transition(Chuyển giao).

* + 1. Pha khởi đầu – Inception: Mô hình hóa nghiệp vụ, hiểu rõ các yêu cầu. Nhằm thu thập các tài liệu như user case, các đặc tả, …
  + 2. Pha xây dựng phát thảo – Elaboration: Hoạt động phân tích và thiết kế. Tạo ra các tài liệu như mô hình thiết kế, giao diện, sự kiện, giao thức, các thiết kế và phân tích của các lớp, thiết kế hệ thống con, mô hình dữ liệu,..
  + 3. Pha xây dựng – Construction: Gồm 2 hoạt động chính là thực hiện và kiểm tra. Thực hiện code hệ thống và kiểm tra sau khi hoàn thành.
  + 4. Pha chuyển giao – Transition: Kế hoạch triển khai, các tài liệu phát hành, tài liệu hỗ trợ, hướng dẫn người dùng cuối, tài liệu cài đặt, đào tạo,..

1. **Trình bày về mô hình kiểm thử hình chữ V, qua đó nêu ngắn gọn các loại kiểm thử cơ bản ứng với từng công đoạn phát triển.**

Mô hình chữ V thực chất là tổ hợp của vòng đời phát triển phần mềm SDLC ở bên trái và vòng đời kiểm thử phần mềm STLC ở bên phải.



* Requirement Analysis (Phân tích yêu cầu) sẽ có quá trình tương ứng là System Testing (Kiểm thử hệ thống): Ở bước này chúng ta sẽ kiểm tra tổng quan toàn bộ hệ thống.
* System Analysis/High Level Design (Thiết kế cấp cao) sẽ có quá trình tương ứng là Integration Testing (Kiểm thử tích hợp): Ở bước này chúng ta sẽ kiểm tra sự kết nối và tương hợp giữa các thành phần của phần mềm.
* Module Design/Low Level Design (Thiết kế cấp thấp) sẽ có quá trình tương ứng là Unit Testing (Kiểm thử đơn vị): Ở bước này chúng ta sẽ kiểm tra ở mức function (tính năng) của phần mềm
* Coding không cần một quá trình kiểm thử tương ứng, thật ra là không cần thiết do ở bước này hầu như các công nghệ và nền tảng kỹ thuật đã hoàn toàn được kiểm tra trước đó bởi nhà sản xuất của mỗi hãng trước khi được sử dụng chính thức. Vì vậy chúng ta không phải kiểm tra lại ở bước này, thường sẽ do Dev đảm bảo.

1. **Trình bày các bước chính trong quy trình kiểm thử phần mềm.**

* Test Planning (Xây dựng kế hoạch kiểm thử): Test Manager hoặc Test Leader sẽ xây dựng kế hoạch ban đầu về kiểm thử (định nghĩa phạm vi kiểm thử, chiến lược kiểm thử, nhận dạng các rủi ro yếu tố bất ngờ,..)
* Test Analyst and Design (Phân tích và thiết kế kiểm thử): Test Analyst hoặc Test Designer sẽ thiết kế (định nghĩa) các testcase từ các yêu cầu liên quan:
  + Thiết kế các testcase từ các yêu cầu chức năng và các yêu cầu không chức năng của phần mềm.
  + Các testcase cần bao phủ tất cả khía cạnh kiểm thử cho từng yêu cầu phần mềm.
  + Các test case cần bao phủ tất cả yêu cầu trong các chiến lược kiểm thử.
  + Nếu cần kiểm thử tự động, Test Designer sẽ xây dựng các kịch bản dựa trên các testcase/Test procedures.
* Test Executing (Thi hành kiểm thử): Testers sẽ được bố trí công việc bởi Test Leader để thi hành kiểm thử.
  + Thi hành kiểm thử theo từng testcase.
  + Thực hiện kiểm thử đặc biệt (ad-hoc)
  + Thực hiện kịch bản kiểm thử mà không được định nghĩa trong testcase.
  + Kiểm thử lại các lỗi đã được sửa chữa.
  + Tester sẽ tạo các báo cáo về lỗi trong suốt quá trình kiểm lỗi và theo dõi chúng cho đến khi chúng đã được xử lý.
  + Ở công đoạn kiểm thử độ chấp thuận, Customer sẽ thi hành kiểm thử để kiểm định xem hệ thống phần mềm có thỏa mãn các nhu cầu người dùng không ?
* Test Report and Evaluation: Test Manager hoặc Test Leader sẽ phân tích các lỗi trong hệ thống theo dõi các lỗi.
  + Tạo các báo cáo lỗi
  + Đánh giá các kết quả kiểm thử, thống kê các yêu cầu thay đổi
  + Tính và phân phối các thông tin đo lường hoạt động kiểm thử
  + Tạo bảng tổng kết đánh giá hoạt động kiểm lỗ
  + Xác định xem đã đạt tiêu chí thành công và hoàn thành kiểm thử chưa

1. **Nêu các nguyên lý cơ bản trong kiểm thử phần mềm.**

* Tiêu chí Đình chỉ (Suspension Criteria): Thử nghiệm sẽ bị tạm dừng nếu các tiêu chí dưới đây xảy ra:
  + Không có ghi chú xây dựng hoặc nó không rõ ràng
  + Có một số lỗi nghiêm trọng trong quá trình tạo thử nghiệm khói mà không có giải pháp khắc phục
* Tiêu chí tiếp tục (Resumption Criteria): Quá trình thử nghiệm sẽ được tiếp tục nếu bản dựng có: Xây dựng ghi chú rõ ràng; bất kỳ lỗi nghiêm trọng nào đối với công việc xung quanh các giải pháp; các trường hợp thử nghiệm đã được căn cứ.
* Thông tin thiết yếu của mỗi testcase là kết quả/dự liêu xuất kỳ vọng. Một testcase phải chứa 2 thành phần thiết yếu: Đặc tả về điều kiện dữ liệu nhập; đặc tả chính xác về kết quả đúng của chương trình tương ứng với dữ liệu nhập.
* Việc kiểm thử đòi hỏi tính độc lập: Lập trình viên nên tránh việc kiểm thử các TPPM do mình viết. Phải thiết kế testcase cho 2 trường hợp: dữ liệu đầu vào hợp lệ và dữ liệu đầu vào không hợp lệ và chờ đợi.
* Xem xét chương trình xem nó không thực hiện những điều mong muốn, xem nó có làm những điều không mong muốn ?
* Tránh các testcase "throwaway" trừ phi chương trình thật sự là "throwaway". Không nên lập kế hoạch nỗ lực kiểm thử dựa trên giả định ngầm rằng phần mềm không có lỗi.
* Xác suất xuất hiện nhiều lỗi hơn trong 1 section phần mềm tỉ lệ thuận với số lỗi đã phát hiện được trong section đó.

1. **Những hạn chế của việc kiểm thử là gì?**

* Không thể chắc là các đặc tả phần mềm đều đúng 100%.
* Không thể chắc rằng hệ thống hay tool kiểm thử là đúng.
* Không có tool kiểm thử nào thích hợp cho mọi phần mềm.
* Kỹ sư kiểm thử không chắc rằng họ hiểu đầy đủ về sản phẩm phần mềm.
* Không bao giờ có đủ tài nguyên để thực hiện kiểm thử đầy đủ phần mềm.
* Không bao giờ chắc rằng ta đạt đủ 100% hoạt động kiểm thử phần mềm.

1. **Test case là gì? Nêu các nguyên tắc cơ bản khi thiết kế test case.**

* Khái niệm: Test case là tập hợp các trường hợp điều kiện mà Tester dựa vào đó để xác định ứng dụng, hệ thống phần mềm hoặc là 1 trong các chức năng của nó có hoạt động như mong muốn hay không.
* Nguyên tắc cơ bản khi thiết kế test case:
  + File testcase cần có những step test đơn giản, minh bạch, dễ hiểu. Step test phải viết chi tiết rõ ràng để ngay cả khi tester khác đọc có thể thực hiện được
  + Mục đích và phạm vi của testcase cũng được mô tả chi tiết. Điều kiện tiền đề, data test cũng phải được ghi ở từng testcase nếu cần.
  + Testcase nên được review chéo bởi member trong team.
  + Không nên gộp quá nhiều kết quả confirm vào 1 case mà nên tách mỗi kết quả confirm ra từng case.
  + Khi tạo testcase nên đứng ở vị trí End use.
  + Testcase nên cover các trường hợp kiểm thử như: Phân lớp tương đương, giá trị biên, điều kiện normal và abnormal.
  + Cả các trường hợp free test không có trong đặc tả yêu cầu.

1. **Trình bày ngắn gọn về các kỹ thuật kiểm thử hộp đen (blackbox testing) và kiểm thử hộp trắng (whitebox testing).**

* ***Blackbox testing****:* là một phương pháp kiểm thử phần mềm mà việc kiểm tra các chức năng của một ứng dụng không cần quan tâm vào cấu trúc nội bộ hoặc hoạt động của nó.Các kỹ thuật sử dụng:
  + Phân vùng tương đương: chia đầu vào thành nhiều vùng giá trị khác nhau mà khi lấy ra một hoặc một vài giá trị trong cùng một vùng thì có kết quả tương đương nhau.
  + Phân tích giá trị biên: à trường hợp đặc biệt của phân vùng tương đương, nó bổ sung thêm cho phân vùng tương đương các trường hợp tại các giá trị biên mà phân vùng tương đương có thể bị sót.
  + Bảng quyết định: là một kỹ thuật tốt với các requirement có nhiều điều kiện đầu vào và các kết quả đầu ra tương ứng
  + Chuyển trạng thái: được áp dụng trong các trường hợp khi mà kết quả đầu ra được kích hoạt bởi sự thay đổi của các điều kiện đầu vào hoặc thay đổi trạng thái của hệ thống.
* ***Whitebox testing:***là phương pháp thử nghiệm phần mềm, trong đó các thiết kế, cấu trúc giải thuật bên trong, và việc thực hiện các công việc đều được biết đến. Các kỹ thuật sử dụng:
  + Kiểm thử đường cơ bản - Đồ thị dòng
  + Kiểm thử dựa trên luồng điều khiển
  + Kiểm thử dựa trên luồng dữ liệu (Data - flow Testing)
  + Kiểm thử đột biến (Mutation Testing).