**Bài tập tổng hợp về Quản lý Testing và Rủi ro**

**Phần 1: Lý thuyết**

**Lập kế hoạch kiểm thử**

**a. Tiêu chí vào/ra (entry/exit criteria) trong kế hoạch kiểm thử là gì? Giải thích tại sao chúng quan trọng.**

**Tiêu chí vào (Entry Criteria):** Đây là các điều kiện cần được đáp ứng trước khi bắt đầu quá trình kiểm thử. Ví dụ: tài liệu yêu cầu (SRS) đã được phê duyệt, môi trường kiểm thử đã được thiết lập, và các test case đã được chuẩn bị.

**Tiêu chí ra (Exit Criteria):** Đây là các điều kiện xác định khi nào quá trình kiểm thử có thể kết thúc. Ví dụ: 95% test case đã pass, các lỗi nghiêm trọng (critical/high severity) đã được khắc phục, và sản phẩm đạt mức chất lượng chấp nhận được.

**Tầm quan trọng:**

* **Tiêu chí vào** đảm bảo rằng kiểm thử bắt đầu trong điều kiện phù hợp, tránh lãng phí thời gian và tài nguyên nếu các điều kiện tiên quyết chưa sẵn sàng.
* **Tiêu chí ra** cung cấp một thước đo rõ ràng để đánh giá xem sản phẩm đã sẵn sàng để phát hành hay chưa, giúp đảm bảo chất lượng và giảm rủi ro.

**b. Cách ưu tiên test case dựa trên mức độ rủi ro và ví dụ cụ thể**

**Cách ưu tiên test case dựa trên rủi ro:**

1. **Xác định rủi ro:** Đánh giá các rủi ro tiềm ẩn dựa trên tác động (impact) và xác suất xảy ra (likelihood). Các tính năng có tác động lớn đến người dùng hoặc doanh nghiệp (ví dụ: thanh toán, bảo mật) được ưu tiên cao.
2. **Phân loại mức độ rủi ro:** Sử dụng ma trận rủi ro (cao, trung bình, thấp) để xếp hạng các tính năng hoặc chức năng.
3. **Ưu tiên test case:** Các test case liên quan đến các tính năng có rủi ro cao được thực hiện trước, đảm bảo rằng các vấn đề nghiêm trọng được phát hiện và khắc phục sớm.

**Ví dụ cụ thể (ứng dụng thương mại điện tử - tính năng thanh toán):**

* **Tính năng:** Thanh toán trực tuyến.
* **Rủi ro:** Lỗi trong tính năng thanh toán có thể dẫn đến mất doanh thu, thất thoát tài chính, hoặc mất lòng tin của khách hàng.
* **Test case ưu tiên cao:**
  1. Kiểm tra thanh toán thành công với các phương thức thanh toán khác nhau (thẻ tín dụng, ví điện tử).
  2. Kiểm tra xử lý lỗi khi nhập sai thông tin thẻ.
  3. Kiểm tra bảo mật giao dịch (mã hóa dữ liệu thanh toán).
* **Lý do ưu tiên:** Thanh toán là tính năng cốt lõi của ứng dụng thương mại điện tử. Lỗi trong tính năng này có thể gây ra thiệt hại tài chính lớn và ảnh hưởng đến uy tín của doanh nghiệp.

**Viết báo cáo defect**

**a. Yếu tố cần có trong báo cáo defect và tầm quan trọng**

Một báo cáo defect cần bao gồm các yếu tố sau:

1. **Tiêu đề defect:** Tóm tắt ngắn gọn vấn đề, giúp đội phát triển dễ dàng nhận diện lỗi.
   * **Tầm quan trọng:** Cung cấp cái nhìn tổng quan, giúp phân loại và ưu tiên xử lý lỗi.
2. **Mô tả defect:** Mô tả chi tiết về lỗi, bao gồm bối cảnh và hành vi không mong muốn.
   * **Tầm quan trọng:** Giúp đội phát triển hiểu rõ bản chất của lỗi.
3. **Bước tái hiện:** Các bước cụ thể để tái tạo lỗi, bao gồm dữ liệu đầu vào và môi trường kiểm thử.
   * **Tầm quan trọng:** Cho phép đội phát triển tái hiện lỗi một cách chính xác để tìm nguyên nhân và sửa lỗi.
4. **Mức độ nghiêm trọng (Severity):** Đánh giá mức độ ảnh hưởng của lỗi (ví dụ: critical, high, medium, low).
   * **Tầm quan trọng:** Giúp đội phát triển ưu tiên sửa các lỗi quan trọng trước.
5. **Kết quả thực tế và kết quả mong đợi:** So sánh giữa hành vi thực tế của hệ thống và hành vi mong muốn.
   * **Tầm quan trọng:** Làm rõ sự khác biệt giữa yêu cầu và thực tế, hỗ trợ việc xác định lỗi.

**b. Ví dụ báo cáo defect và vai trò hỗ trợ đội phát triển**

**Ví dụ báo cáo defect (lỗi hiển thị sai giá sản phẩm):**

* **Tiêu đề defect:** Giá sản phẩm hiển thị sai trong giỏ hàng.
* **Mô tả defect:** Khi thêm sản phẩm vào giỏ hàng, giá hiển thị trong giỏ không khớp với giá trên trang chi tiết sản phẩm.
* **Bước tái hiện:**
  1. Truy cập trang web thương mại điện tử.
  2. Chọn sản phẩm "Tai nghe Bluetooth" với giá 500,000 VNĐ.
  3. Thêm sản phẩm vào giỏ hàng.
  4. Kiểm tra giá trong giỏ hàng.
* **Mức độ nghiêm trọng:** Cao (gây nhầm lẫn cho khách hàng và có thể dẫn đến khiếu nại).
* **Kết quả thực tế:** Giá trong giỏ hàng hiển thị là 450,000 VNĐ.
* **Kết quả mong đợi:** Giá trong giỏ hàng phải hiển thị đúng là 500,000 VNĐ.

**Cách báo cáo này hỗ trợ đội phát triển:**

* **Tái hiện lỗi:** Các bước tái hiện rõ ràng giúp lập trình viên nhanh chóng xác định vị trí lỗi trong mã nguồn.
* **Ưu tiên sửa lỗi:** Mức độ nghiêm trọng cao thúc đẩy đội phát triển xử lý lỗi nhanh chóng.
* **Hiểu rõ yêu cầu:** Kết quả mong đợi giúp lập trình viên biết chính xác cần sửa gì để đáp ứng yêu cầu.

**Phân tích rủi ro sản phẩm và dự án**

**a. Phân biệt rủi ro sản phẩm và rủi ro dự án**

* **Rủi ro sản phẩm:** Liên quan đến chất lượng hoặc chức năng của sản phẩm, ảnh hưởng trực tiếp đến người dùng cuối.
  + **Ví dụ:** Lỗi trong tính năng xác thực người dùng có thể dẫn đến vi phạm bảo mật.
* **Rủi ro dự án:** Liên quan đến quá trình phát triển hoặc kiểm thử, ảnh hưởng đến tiến độ, chi phí, hoặc tài nguyên của dự án.
  + **Ví dụ:** Thiếu nhân sự kiểm thử dẫn đến việc không hoàn thành kiểm thử đúng hạn.

**b. Cách phân tích rủi ro để tập trung nỗ lực kiểm thử**

1. **Xác định rủi ro:** Liệt kê các rủi ro tiềm ẩn dựa trên yêu cầu, chức năng, và bối cảnh dự án.
2. **Đánh giá rủi ro:** Sử dụng ma trận rủi ro (tác động x xác suất) để phân loại rủi ro.
3. **Tập trung kiểm thử:** Ưu tiên kiểm thử các khu vực có rủi ro cao, chẳng hạn như các tính năng cốt lõi, các chức năng phức tạp, hoặc các khu vực có lịch sử lỗi nhiều.
4. **Giảm thiểu rủi ro:** Áp dụng các chiến lược như kiểm thử tự động cho các trường hợp lặp lại, hoặc tăng cường kiểm thử bảo mật cho các khu vực nhạy cảm.

**Ví dụ:** Trong một ứng dụng ngân hàng trực tuyến, tính năng chuyển tiền có rủi ro cao (tác động lớn đến tài chính và uy tín). Do đó, cần tập trung kiểm thử các trường hợp như chuyển tiền không thành công, sai tài khoản, hoặc lỗi bảo mật.

**Test Plan và theo dõi tiến độ kiểm thử**

**a. Thành phần của một test plan cơ bản**

Một test plan cơ bản bao gồm:

1. **Mục tiêu kiểm thử:** Xác định mục đích của kiểm thử, ví dụ: đảm bảo sản phẩm đáp ứng yêu cầu chức năng và phi chức năng.
2. **Phạm vi kiểm thử:** Xác định các tính năng sẽ được kiểm thử và không được kiểm thử.
3. **Tài nguyên:** Liệt kê nhân sự, công cụ, và môi trường kiểm thử cần thiết.
4. **Lịch trình kiểm thử:** Thời gian bắt đầu, kết thúc, và các mốc quan trọng.
5. **Tiêu chí vào/ra:** Điều kiện để bắt đầu và kết thúc kiểm thử.

**b. Sử dụng các chỉ số để theo dõi tiến độ kiểm thử và đánh giá chất lượng**

* **Tỷ lệ test case hoàn thành:** Đo lường số lượng test case đã thực hiện so với tổng số test case. Ví dụ: 80/100 test case hoàn thành (80%) cho thấy tiến độ kiểm thử.
* **Số lượng defect:** Theo dõi số lượng lỗi được phát hiện, phân loại theo mức độ nghiêm trọng (critical, high, medium, low). Ví dụ: 10 lỗi critical cần được ưu tiên sửa.
* **Tỷ lệ defect được giải quyết:** Đo lường số lượng lỗi đã được khắc phục so với tổng số lỗi. Ví dụ: 90% lỗi critical đã được giải quyết cho thấy chất lượng cải thiện.
* **Tầm quan trọng:** Các chỉ số này giúp đánh giá tiến độ kiểm thử, xác định các khu vực cần tập trung, và đảm bảo sản phẩm đạt chất lượng trước khi phát hành.

**Phần 2: Bài tập thực hành**

**Lập kế hoạch kiểm thử (Ứng dụng đặt lịch khám bác sĩ trực tuyến)**

**a. Tiêu chí vào/ra**

* **Tiêu chí vào:**
  1. Tài liệu yêu cầu (SRS) đã được phê duyệt.
  2. Môi trường kiểm thử (máy chủ, cơ sở dữ liệu, thiết bị) đã được thiết lập.
  3. Test case và dữ liệu kiểm thử đã được chuẩn bị và phê duyệt.
* **Tiêu chí ra:**
  1. 95% test case pass.
  2. Tất cả các lỗi critical và high severity đã được khắc phục.
  3. Hệ thống đáp ứng các yêu cầu chức năng và phi chức năng (hiệu suất, bảo mật).

**b. Test case ưu tiên cao dựa trên rủi ro**

1. **Test case:** Kiểm tra tính năng xác nhận lịch hẹn.
   * **Lý do ưu tiên:** Đây là tính năng cốt lõi, nếu lỗi xảy ra (ví dụ: xác nhận sai lịch hẹn), bệnh nhân có thể bỏ lỡ cuộc hẹn, gây ảnh hưởng đến sức khỏe và uy tín của ứng dụng.
2. **Test case:** Kiểm tra tính năng thông báo lịch hẹn qua email/SMS.
   * **Lý do ưu tiên:** Lỗi trong thông báo có thể dẫn đến việc bệnh nhân không nhận được thông tin, gây nhầm lẫn và khiếu nại.
3. **Test case:** Kiểm tra bảo mật khi đăng nhập và truy cập thông tin bệnh nhân.
   * **Lý do ưu tiên:** Lỗi bảo mật có thể dẫn đến rò rỉ dữ liệu nhạy cảm, vi phạm quy định bảo mật (như GDPR) và gây mất lòng tin.

**Viết báo cáo defect (Lỗi đặt lịch vào giờ nghỉ trưa)**

**a. Báo cáo defect chi tiết**

* **Tiêu đề defect:** Hệ thống cho phép đặt lịch khám vào giờ nghỉ trưa của bác sĩ (12:00-13:00).
* **Mô tả defect:** Hệ thống cho phép người dùng đặt lịch khám trong khung giờ nghỉ trưa của bác sĩ (12:00-13:00), dẫn đến lịch hẹn không hợp lệ.
* **Bước tái hiện:**
  1. Đăng nhập vào ứng dụng với tài khoản người dùng.
  2. Chọn bác sĩ và ngày khám (ví dụ: 05/07/2025).
  3. Chọn khung giờ 12:30 và gửi yêu cầu đặt lịch.
  4. Hệ thống xác nhận lịch hẹn thành công.
* **Mức độ nghiêm trọng:** Cao (gây ra lịch hẹn không hợp lệ, ảnh hưởng đến trải nghiệm người dùng và lịch trình bác sĩ).
* **Kết quả thực tế:** Hệ thống cho phép đặt lịch vào 12:30 và xác nhận thành công.
* **Kết quả mong đợi:** Hệ thống hiển thị thông báo lỗi, không cho phép đặt lịch trong khung giờ 12:00-13:00.

**b. Cách báo cáo này giúp đội phát triển**

* **Tái hiện lỗi:** Các bước tái hiện rõ ràng giúp lập trình viên xác định đoạn mã xử lý logic kiểm tra khung giờ.
* **Hiểu rõ vấn đề:** Mô tả và kết quả mong đợi chỉ ra rằng cần thêm kiểm tra khung giờ nghỉ trong quy trình đặt lịch.
* **Ưu tiên sửa lỗi:** Mức độ nghiêm trọng cao đảm bảo lỗi được xử lý ngay lập tức, tránh ảnh hưởng đến lịch trình bác sĩ và trải nghiệm người dùng.

**Phân tích rủi ro và soạn Test Plan (Ứng dụng quản lý kho hàng)**

**a. Phân tích rủi ro**

**Rủi ro sản phẩm:**

1. **Lỗi tính toán số lượng tồn kho:** Nếu số lượng tồn kho hiển thị sai, có thể dẫn đến đặt hàng quá mức hoặc thiếu hàng, gây thiệt hại tài chính.
2. **Lỗi nhập/xuất kho:** Lỗi trong quy trình nhập/xuất kho có thể gây ra sai lệch dữ liệu, ảnh hưởng đến hoạt động kinh doanh.
3. **Lỗi bảo mật dữ liệu kho:** Rò rỉ thông tin kho hàng (ví dụ: giá nhập hàng) có thể gây tổn thất cạnh tranh.

**Rủi ro dự án:**

1. **Thiếu tài nguyên kiểm thử:** Số lượng nhân sự kiểm thử hạn chế có thể dẫn đến kiểm thử không đầy đủ, bỏ sót lỗi nghiêm trọng.
2. **Trễ tiến độ phát triển:** Nếu nhóm phát triển không hoàn thành đúng hạn, thời gian kiểm thử sẽ bị rút ngắn, làm giảm chất lượng.

**b. Test Plan cơ bản**

**Mục tiêu kiểm thử:**

* Đảm bảo ứng dụng quản lý kho hàng đáp ứng các yêu cầu chức năng (nhập/xuất kho, tính toán tồn kho) và phi chức năng (hiệu suất, bảo mật).
* Phát hiện và khắc phục các lỗi nghiêm trọng trước khi phát hành.

**Phạm vi kiểm thử:**

* **Trong phạm vi:** Kiểm thử các tính năng nhập/xuất kho, báo cáo tồn kho, quản lý danh mục sản phẩm, và bảo mật dữ liệu.
* **Ngoài phạm vi:** Kiểm thử tích hợp với phần mềm bên thứ ba chưa được triển khai.

**Tài nguyên:**

* **Nhân sự:** 1 Test Lead, 3 Tester.
* **Công cụ:** Selenium (kiểm thử tự động), Jira (quản lý defect), TestRail (quản lý test case).
* **Môi trường kiểm thử:** Máy chủ staging, cơ sở dữ liệu MySQL, trình duyệt Chrome/Firefox.

**Lịch trình kiểm thử:**

* Tuần 1: Chuẩn bị test case, thiết lập môi trường kiểm thử.
* Tuần 2-3: Thực hiện kiểm thử chức năng và phi chức năng.
* Tuần 4: Kiểm thử hồi quy và báo cáo kết quả.

**Chỉ số theo dõi tiến độ:**

* Tỷ lệ test case hoàn thành: Mục tiêu 100% test case được thực hiện.
* Số lượng defect: Theo dõi số lượng lỗi theo mức độ nghiêm trọng.
* Tỷ lệ defect được giải quyết: Mục tiêu 100% lỗi critical/high được khắc phục.
* Độ bao phủ kiểm thử: Mục tiêu bao phủ 95% yêu cầu chức năng.