## **Phần 1: Lý thuyết**

### **Nguyên tắc cơ bản của Testing**

#### **a. Liệt kê và tóm tắt 7 nguyên tắc cơ bản của testing**

1. **Testing cho thấy sự hiện diện của lỗi, không phải sự vắng mặt của lỗi**
   * Tóm tắt: Testing giúp phát hiện lỗi nhưng không đảm bảo phần mềm không có lỗi.
   * Ví dụ: Kiểm tra chức năng đăng nhập của ứng dụng, phát hiện lỗi khi nhập sai mật khẩu, nhưng không thể khẳng định hệ thống hoàn toàn không có lỗi khác.
2. **Kiểm thử toàn diện là không thể**
   * Tóm tắt: Không thể kiểm tra mọi trường hợp, cần ưu tiên các trường hợp quan trọng.
   * Ví dụ: Trong ứng dụng thương mại điện tử, không thể kiểm tra mọi kết hợp sản phẩm trong giỏ hàng, nên tập trung vào các trường hợp phổ biến như thêm, xóa sản phẩm.
3. **Kiểm thử sớm**
   * Tóm tắt: Bắt đầu kiểm thử từ giai đoạn đầu của dự án để giảm chi phí sửa lỗi.
   * Ví dụ: Kiểm tra yêu cầu của ứng dụng trước khi phát triển để phát hiện mâu thuẫn trong yêu cầu người dùng.
4. **Tập hợp lỗi**
   * Tóm tắt: Lỗi thường tập trung ở một số module nhất định.
   * Ví dụ: Trong ứng dụng di động, chức năng thanh toán thường có nhiều lỗi hơn các chức năng khác như hiển thị thông tin sản phẩm.
5. **Nghịch lý thuốc trừ sâu**
   * Tóm tắt: Lặp lại cùng bộ kiểm thử sẽ giảm hiệu quả vì không phát hiện lỗi mới.
   * Ví dụ: Nếu chỉ kiểm tra chức năng đăng nhập với cùng bộ dữ liệu, các lỗi tiềm ẩn trong các trường hợp khác (như tài khoản bị khóa) sẽ không được phát hiện.
6. **Testing phụ thuộc vào ngữ cảnh**
   * Tóm tắt: Quy trình kiểm thử phải phù hợp với loại dự án và yêu cầu.
   * Ví dụ: Một ứng dụng ngân hàng yêu cầu kiểm thử bảo mật nghiêm ngặt hơn so với một trò chơi di động.
7. **Không có lỗi không có nghĩa là sản phẩm tốt**
   * Tóm tắt: Sản phẩm không có lỗi vẫn có thể không đáp ứng nhu cầu người dùng.
   * Ví dụ: Một ứng dụng chạy ổn định nhưng giao diện khó sử dụng sẽ không được người dùng chấp nhận.

#### **b. Áp dụng nguyên tắc vào tình huống thực tế**

**Nguyên tắc: Kiểm thử sớm**

* **Tình huống**: Trong một ứng dụng thương mại điện tử, việc kiểm thử sớm được áp dụng bằng cách xem xét tài liệu yêu cầu (requirements) trước khi bắt đầu coding.
* **Cách áp dụng**: Nhóm kiểm thử phối hợp với nhóm phân tích yêu cầu để xác định các trường hợp sử dụng chính, như thêm sản phẩm vào giỏ hàng, thanh toán, hoặc hủy đơn hàng. Họ phát hiện một yêu cầu mâu thuẫn: hệ thống cho phép thêm số lượng sản phẩm âm. Bằng cách phát hiện sớm, nhóm phát triển sửa đổi yêu cầu trước khi lập trình, tiết kiệm thời gian và chi phí sửa lỗi sau này.

### **Hoạt động Testing và Testware**

#### **a. Các hoạt động chính trong quy trình testing**

1. **Lập kế hoạch kiểm thử**: Xác định mục tiêu, phạm vi, tài nguyên, và lịch trình kiểm thử.
2. **Thiết kế kiểm thử**: Xây dựng các kịch bản kiểm thử (test cases) và dữ liệu kiểm thử dựa trên yêu cầu.
3. **Thực thi kiểm thử**: Chạy các kịch bản kiểm thử, ghi nhận kết quả và lỗi.
4. **Báo cáo kiểm thử**: Tổng hợp kết quả, báo cáo lỗi, và đề xuất cải tiến.

#### **b. Testware là gì?**

* **Định nghĩa**: Testware là tập hợp các tài liệu, công cụ, và dữ liệu được sử dụng trong quá trình kiểm thử để đảm bảo chất lượng phần mềm.
* **3 loại testware**:
  1. **Test Plan**: Tài liệu mô tả chiến lược, mục tiêu, và phạm vi kiểm thử. Vai trò: Định hướng toàn bộ quá trình kiểm thử.
  2. **Test Case**: Tài liệu chi tiết các bước kiểm thử, dữ liệu đầu vào, và kết quả mong đợi. Vai trò: Hướng dẫn tester thực hiện kiểm thử một cách nhất quán.
  3. **Test Data**: Dữ liệu được tạo để sử dụng trong kiểm thử. Vai trò: Mô phỏng các tình huống thực tế để kiểm tra hành vi hệ thống.

#### **c. Ví dụ sử dụng testware trong dự án**

* **Dự án**: Ứng dụng di động đặt vé xem phim.
* **Ví dụ**:
  + **Test Plan**: Xác định kiểm thử chức năng đặt vé, thanh toán, và hủy vé trên các hệ điều hành (iOS, Android).
  + **Test Case**: Kiểm tra chức năng chọn ghế với các trường hợp như chọn ghế đã đặt, chọn nhiều ghế, hoặc hủy chọn.
  + **Test Data**: Tạo dữ liệu như tài khoản người dùng giả lập, danh sách phim, và các ghế trống/đã đặt để kiểm tra.

#### **Vai trò của các thành viên trong đội kiểm thử**

##### **a. Các vai trò chính**

1. **Test Manager**:
   * **Trách nhiệm**: Lập kế hoạch kiểm thử, phân bổ tài nguyên, giám sát tiến độ, và báo cáo kết quả cho các bên liên quan.
2. **Test Analyst**:
   * **Trách nhiệm**: Phân tích yêu cầu, thiết kế test case, và xác định các kịch bản kiểm thử phức tạp.
3. **Tester**:
   * **Trách nhiệm**: Thực thi các test case, báo cáo lỗi, và kiểm tra lại sau khi lỗi được sửa.

##### **b. Phối hợp giữa các vai trò**

* **Cách phối hợp**:
  + Test Manager xây dựng kế hoạch kiểm thử và phân công nhiệm vụ.
  + Test Analyst thiết kế test case dựa trên yêu cầu và chia sẻ với Tester.
  + Tester thực thi kiểm thử, ghi nhận lỗi, và phối hợp với Test Analyst để xác minh lỗi.
  + Test Manager tổng hợp báo cáo và làm việc với nhóm phát triển để đảm bảo lỗi được sửa trước khi phát hành.
* **Ví dụ**: Trong dự án ứng dụng thương mại điện tử, Test Manager xác định phạm vi kiểm thử (chức năng giỏ hàng, thanh toán). Test Analyst tạo test case cho các trường hợp như thêm sản phẩm hết hàng. Tester chạy test case và báo cáo lỗi (hệ thống cho phép thêm sản phẩm không còn hàng). Test Manager làm việc với nhóm phát triển để sửa lỗi và lên kế hoạch kiểm thử lại.

#### **Điều chỉnh quy trình Testing theo ngữ cảnh dự án**

##### **a. Tại sao cần điều chỉnh và ví dụ**

* **Lý do**: Mỗi dự án có yêu cầu, rủi ro, và tài nguyên khác nhau, do đó quy trình kiểm thử cần được điều chỉnh để phù hợp.
* **Ví dụ**:
  1. **Ứng dụng ngân hàng**:
     + **Quy trình**: Tập trung vào kiểm thử bảo mật (xác thực, mã hóa dữ liệu) và kiểm thử hiệu năng (xử lý giao dịch lớn). Sử dụng kiểm thử tự động để kiểm tra các giao dịch lặp lại.
  2. **Trò chơi di động**:
     + **Quy trình**: Ưu tiên kiểm thử trải nghiệm người dùng (UX) và tương thích trên nhiều thiết bị. Sử dụng kiểm thử thủ công để đánh giá giao diện và hiệu ứng đồ họa.

##### **b. Yếu tố cần xem xét**

1. **Quy mô dự án**: Dự án lớn cần quy trình kiểm thử chi tiết hơn, với nhiều test case và công cụ tự động hóa.
2. **Mức độ rủi ro**: Dự án có rủi ro cao (như ứng dụng y tế) cần kiểm thử toàn diện hơn, đặc biệt về bảo mật và độ tin cậy.
3. **Thời gian**: Dự án có thời gian gấp rút có thể ưu tiên kiểm thử các chức năng chính và sử dụng công cụ tự động để tiết kiệm thời gian.

## **Phần 2: Bài tập thực hành**

### **Áp dụng nguyên tắc Testing**

**Ứng dụng đặt đồ ăn trực tuyến – Tính năng "thêm món ăn vào giỏ hàng"**

1. **Nguyên tắc: Testing cho thấy sự hiện diện của lỗi**
   * **Áp dụng**: Kiểm tra trường hợp người dùng nhập số lượng món ăn âm, số lượng vượt quá giới hạn (ví dụ: 100 món), hoặc thêm món ăn đã hết hàng. Phát hiện lỗi nếu hệ thống cho phép thêm trong các trường hợp này.
2. **Nguyên tắc: Kiểm thử sớm**
   * **Áp dụng**: Xem xét yêu cầu tính năng "thêm món ăn" trước khi phát triển để đảm bảo hệ thống giới hạn số lượng món ăn hợp lệ (ví dụ: từ 1 đến 10) và hiển thị thông báo lỗi rõ ràng khi vi phạm.
3. **Nguyên tắc: Testing phụ thuộc vào ngữ cảnh**
   * **Áp dụng**: Vì ứng dụng đặt đồ ăn ưu tiên trải nghiệm người dùng, kiểm thử tập trung vào giao diện (hiển thị thông báo khi thêm món thành công/thất bại) và tương thích trên các thiết bị di động (iOS, Android).

### **Thiết kế quy trình Testing**

**Quy trình testing cho ứng dụng đặt đồ ăn trực tuyến**

1. **Lập kế hoạch kiểm thử**
   * **Bước thực hiện**: Xác định phạm vi (chức năng thêm món, thanh toán, giao hàng), tài nguyên (tester, công cụ), và lịch trình.
   * **Testware**: Test Plan (mô tả mục tiêu, phạm vi, và lịch trình).
2. **Thiết kế kiểm thử**
   * **Bước thực hiện**: Tạo test case cho các trường hợp như thêm món hợp lệ, thêm món hết hàng, thêm số lượng không hợp lệ. Chuẩn bị dữ liệu kiểm thử (danh sách món ăn, số lượng).
   * **Testware**: Test Case, Test Data.
3. **Thực thi kiểm thử**
   * **Bước thực hiện**: Chạy test case trên các thiết bị (iOS, Android), ghi nhận lỗi (ví dụ: hệ thống không thông báo khi thêm món hết hàng).
   * **Testware**: Test Log (ghi lại kết quả thực thi).
4. **Báo cáo kiểm thử**
   * **Bước thực hiện**: Tổng hợp kết quả, báo cáo lỗi, và đề xuất cải tiến (ví dụ: thêm thông báo lỗi rõ ràng).
   * **Testware**: Test Report (tóm tắt kết quả và lỗi).

### **Phân tích ngữ cảnh dự án**

#### **a. Yếu tố ngữ cảnh ảnh hưởng đến quy trình testing**

1. **Hệ thống quản lý bệnh viện**
   * **Yếu tố**:
     + **Yêu cầu bảo mật**: Dữ liệu bệnh nhân phải được mã hóa và tuân thủ quy định như HIPAA.
     + **Độ tin cậy**: Hệ thống phải hoạt động liên tục, không được phép có lỗi nghiêm trọng.
   * **Quy trình**: Ưu tiên kiểm thử bảo mật (xâm nhập, xác thực) và kiểm thử tải (đáp ứng khi có nhiều người dùng cùng truy cập).
2. **Ứng dụng chỉnh sửa ảnh trên di động**
   * **Yếu tố**:
     + **Trải nghiệm người dùng**: Giao diện phải thân thiện, dễ sử dụng.
     + **Tương thích thiết bị**: Hỗ trợ nhiều phiên bản hệ điều hành và kích thước màn hình.
   * **Quy trình**: Tập trung vào kiểm thử UX (giao diện, hiệu ứng) và kiểm thử tương thích trên các thiết bị.

#### **b. Đề xuất điều chỉnh quy trình testing**

1. **Hệ thống quản lý bệnh viện**
   * **Điều chỉnh**:
     + Sử dụng kiểm thử tự động để kiểm tra bảo mật (ví dụ: quét lỗ hổng bảo mật).
     + Tăng cường kiểm thử tích hợp để đảm bảo các module (quản lý bệnh nhân, lịch hẹn) hoạt động tốt với nhau.
     + Ưu tiên kiểm thử các trường hợp rủi ro cao như mất dữ liệu bệnh nhân.
2. **Ứng dụng chỉnh sửa ảnh trên di động**
   * **Điều chỉnh**:
     + Sử dụng kiểm thử thủ công để đánh giá giao diện và trải nghiệm người dùng.
     + Kiểm thử tương thích trên nhiều thiết bị và phiên bản hệ điều hành (ví dụ: iOS 16, Android 12).
     + Tối ưu hóa kiểm thử hiệu năng để đảm bảo ứng dụng chạy mượt mà khi áp dụng hiệu ứng phức tạp.