### **Phần 1: Lý thuyết**

#### **Giá trị của Static Testing**

**a. Static Testing là gì? Giải thích giá trị của static testing trong việc phát hiện lỗi trước khi chạy phần mềm.**

Static Testing là quá trình kiểm tra các tài liệu, mã nguồn hoặc sản phẩm phần mềm mà không thực thi chương trình. Nó bao gồm các hoạt động như review tài liệu, phân tích mã nguồn, và kiểm tra thiết kế để phát hiện lỗi sớm trong chu kỳ phát triển phần mềm.

**Giá trị của Static Testing:**

* **Phát hiện lỗi sớm**: Static Testing giúp phát hiện lỗi trong giai đoạn đầu (như yêu cầu, thiết kế), khi chi phí sửa lỗi còn thấp. Theo nghiên cứu, chi phí sửa lỗi tăng gấp 10 lần khi lỗi được phát hiện ở giai đoạn sau (như kiểm thử hoặc triển khai).
* **Cải thiện chất lượng**: Phát hiện các vấn đề như yêu cầu mâu thuẫn, thiếu rõ ràng, hoặc lỗi logic trước khi phát triển, từ đó giảm thiểu rủi ro trong các giai đoạn sau.
* **Tiết kiệm thời gian và tài nguyên**: Giảm số lượng lỗi cần sửa trong giai đoạn kiểm thử động (dynamic testing), giúp tiết kiệm thời gian và công sức.
* **Tăng cường hiểu biết chung**: Các hoạt động review trong static testing giúp đội ngũ hiểu rõ hơn về yêu cầu và thiết kế, cải thiện giao tiếp và phối hợp.

**b. Ví dụ về lỗi phát hiện qua Static Testing và tầm quan trọng của việc phát hiện sớm.**

**Ví dụ**: Trong tài liệu yêu cầu, một yêu cầu ghi: “Hệ thống phải xử lý 100 giao dịch mỗi giây” nhưng không nêu rõ điều kiện (ví dụ: trong môi trường mạng ổn định hay có tải cao). Lỗi này có thể dẫn đến việc phát triển hệ thống không đáp ứng được trong các tình huống thực tế.

**Tầm quan trọng của phát hiện sớm**:

* Nếu lỗi này không được phát hiện sớm, nhóm phát triển có thể xây dựng hệ thống dựa trên giả định sai, dẫn đến hệ thống không đáp ứng được yêu cầu thực tế.
* Việc sửa lỗi ở giai đoạn sau (như sau khi triển khai) có thể yêu cầu thay đổi mã nguồn, kiểm thử lại, và thậm chí tái cấu trúc hệ thống, gây tốn kém thời gian và chi phí.
* Phát hiện sớm giúp làm rõ yêu cầu ngay từ đầu, đảm bảo hệ thống được thiết kế đúng ngay từ giai đoạn ban đầu.

#### **Các loại tài liệu hoặc sản phẩm trong Static Testing**

**a. Liệt kê ít nhất 5 loại tài liệu hoặc sản phẩm có thể được kiểm tra bằng Static Testing.**

1. **Tài liệu yêu cầu phần mềm (Software Requirements Specification - SRS)**.
2. **Tài liệu thiết kế hệ thống (System Design Document)**.
3. **Mã nguồn (Source Code)**.
4. **Tài liệu kiểm thử (Test Plan/Test Cases)**.
5. **Thiết kế giao diện người dùng (UI Mockups/Wireframes)**.

**b. Ví dụ lỗi tiềm ẩn cho mỗi loại tài liệu/sản phẩm.**

1. **Tài liệu yêu cầu phần mềm**: Yêu cầu mâu thuẫn, ví dụ: “Hệ thống cho phép xóa tài khoản người dùng” và “Tài khoản người dùng không bao giờ được xóa”.
   * **Tác động**: Gây nhầm lẫn cho nhóm phát triển, dẫn đến lỗi trong triển khai.
2. **Tài liệu thiết kế hệ thống**: Thiếu mô tả về cách xử lý lỗi (error handling) trong trường hợp mất kết nối mạng.
   * **Tác động**: Hệ thống có thể gặp lỗi không mong muốn khi triển khai thực tế.
3. **Mã nguồn**: Biến không được khởi tạo trước khi sử dụng trong mã nguồn.
   * **Tác động**: Gây lỗi runtime (lỗi khi chạy chương trình) hoặc hành vi không xác định.
4. **Tài liệu kiểm thử**: Test case không bao quát các trường hợp biên (boundary cases), ví dụ: không kiểm tra trường hợp nhập số âm cho tuổi người dùng.
   * **Tác động**: Lỗi liên quan đến trường hợp biên không được phát hiện trong kiểm thử.
5. **Thiết kế giao diện người dùng**: Thiếu nút “Hủy” trong form nhập liệu.
   * **Tác động**: Gây khó khăn cho người dùng, làm giảm trải nghiệm người dùng (UX).

#### **Quy trình Review**

**a. Các bước chính trong quy trình review.**

1. **Lập kế hoạch**: Xác định mục tiêu review, tài liệu/sản phẩm cần review, loại review (walkthrough, inspection), và phân công vai trò (tác giả, người review, người điều phối).
2. **Chuẩn bị**: Người review đọc trước tài liệu, ghi chú các vấn đề tiềm ẩn, và chuẩn bị câu hỏi hoặc nhận xét.
3. **Thực hiện review**: Tổ chức cuộc họp review (nếu cần), thảo luận các vấn đề, ghi nhận lỗi và đề xuất cải tiến.
4. **Báo cáo và sửa lỗi**: Tổng hợp các lỗi được phát hiện, phân loại mức độ nghiêm trọng, và giao cho tác giả để sửa chữa.
5. **Kiểm tra lại (Follow-up)**: Đảm bảo các lỗi đã được sửa và tài liệu/sản phẩm đạt chất lượng mong muốn.

**b. Vai trò trong quy trình review và trách nhiệm.**

1. **Tác giả (Author)**:
   * **Trách nhiệm**: Chuẩn bị tài liệu/sản phẩm cần review, giải thích nội dung khi cần, và sửa chữa các lỗi được phát hiện.
2. **Người review (Reviewer)**:
   * **Trách nhiệm**: Đọc và phân tích tài liệu, tìm kiếm lỗi, mâu thuẫn hoặc thiếu sót, và đưa ra phản hồi mang tính xây dựng.
3. **Người điều phối (Moderator)**:
   * **Trách nhiệm**: Quản lý quy trình review, đảm bảo cuộc họp diễn ra hiệu quả, giải quyết xung đột, và theo dõi tiến độ sửa lỗi.

**c. Cách chọn loại review phù hợp.**

* **Walkthrough**: Phù hợp khi mục tiêu là chia sẻ kiến thức hoặc thu thập ý kiến ban đầu. Ví dụ: Review tài liệu yêu cầu mới với nhóm để đảm bảo tất cả hiểu rõ nội dung.
  + **Lý do chọn**: Ít trang trọng, tiết kiệm thời gian, phù hợp với các dự án nhỏ hoặc giai đoạn đầu.
* **Inspection**: Phù hợp khi cần kiểm tra kỹ lưỡng để đảm bảo chất lượng cao, ví dụ: Review mã nguồn trước khi tích hợp vào hệ thống chính.
  + **Lý do chọn**: Trang trọng, có quy trình rõ ràng, phù hợp với các dự án lớn hoặc có yêu cầu nghiêm ngặt về chất lượng.
* **Informal Review**: Phù hợp khi cần phản hồi nhanh từ một số ít người, ví dụ: Kiểm tra nhanh tài liệu thiết kế giao diện trước khi gửi khách hàng.
  + **Lý do chọn**: Nhanh chóng, ít tốn kém, phù hợp với các tài liệu không quá phức tạp.

**Tiêu chí lựa chọn**: Dựa trên mục tiêu (chất lượng cao hay chia sẻ thông tin), độ phức tạp của tài liệu, thời gian và nguồn lực sẵn có.

#### **Phản hồi từ Stakeholder**

**a. Tại sao phản hồi thường xuyên từ stakeholder giúp cải thiện chất lượng tài liệu và sản phẩm.**

* **Đảm bảo đáp ứng nhu cầu thực tế**: Stakeholder (như khách hàng, người dùng cuối) cung cấp thông tin về yêu cầu thực tế, giúp tài liệu/sản phẩm sát với mong đợi.
* **Phát hiện vấn đề sớm**: Phản hồi giúp phát hiện các thiếu sót hoặc hiểu lầm trong tài liệu trước khi phát triển, giảm chi phí sửa lỗi.
* **Cải thiện giao tiếp**: Phản hồi thường xuyên tạo kênh giao tiếp liên tục giữa nhóm phát triển và stakeholder, đảm bảo sự đồng thuận.
* **Tăng tính khả thi**: Phản hồi giúp điều chỉnh các yêu cầu không thực tế hoặc khó triển khai.

**b. Ví dụ về phản hồi từ stakeholder cải thiện tài liệu.**

**Ví dụ**: Trong tài liệu yêu cầu phần mềm cho một ứng dụng đặt vé, yêu cầu ban đầu ghi: “Hệ thống hiển thị tất cả các sự kiện có sẵn.” Trong quá trình review, stakeholder (người tổ chức sự kiện) phản hồi rằng cần có bộ lọc theo loại sự kiện (âm nhạc, thể thao) để người dùng dễ tìm kiếm.

**Cải thiện**: Yêu cầu được sửa đổi thành: “Hệ thống hiển thị tất cả các sự kiện có sẵn với bộ lọc theo loại sự kiện (âm nhạc, thể thao, hội thảo).” Điều này làm rõ yêu cầu, cải thiện trải nghiệm người dùng, và tránh việc phát triển một giao diện không đáp ứng nhu cầu thực tế.

### **Phần 2: Bài tập thực hành**

#### **Áp dụng Static Testing**

**Yêu cầu**: “Hệ thống cho phép người dùng đăng ký sự kiện mà không cần kiểm tra số lượng chỗ trống.”

**a. Xác định ít nhất 2 lỗi tiềm năng.**

1. **Thiếu kiểm tra số lượng chỗ trống**:
   * **Lỗi**: Yêu cầu cho phép đăng ký không giới hạn, dẫn đến tình trạng đăng ký vượt quá số chỗ trống, gây ra lỗi hệ thống hoặc bất mãn từ người dùng.
   * **Tác động**: Người dùng có thể đăng ký nhưng không được tham gia sự kiện do hết chỗ, làm giảm uy tín của hệ thống.
2. **Thiếu rõ ràng về trạng thái đăng ký**:
   * **Lỗi**: Yêu cầu không nêu rõ hệ thống sẽ xử lý thế nào khi đăng ký thành công (ví dụ: gửi email xác nhận, cập nhật trạng thái sự kiện).
   * **Tác động**: Gây nhầm lẫn cho người dùng và nhóm phát triển, dẫn đến trải nghiệm người dùng kém hoặc lỗi trong triển khai.

**b. Đề xuất sửa đổi yêu cầu và giải thích.**

**Sửa đổi**:

* **Yêu cầu sửa đổi**: “Hệ thống cho phép người dùng đăng ký sự kiện chỉ khi còn chỗ trống. Hệ thống sẽ kiểm tra số lượng chỗ trống trước khi cho phép đăng ký và gửi email xác nhận khi đăng ký thành công.”
* **Giải thích cải thiện**:
  + **Kiểm tra số lượng chỗ trống**: Đảm bảo hệ thống không cho phép đăng ký quá số lượng, tránh lỗi vận hành và cải thiện trải nghiệm người dùng.
  + **Thêm xác nhận đăng ký**: Làm rõ hành vi của hệ thống sau đăng ký, giúp người dùng biết trạng thái đăng ký và tăng tính minh bạch.
  + **Rõ ràng hơn**: Giúp nhóm phát triển hiểu chính xác yêu cầu, giảm nguy cơ hiểu lầm hoặc triển khai sai.

#### **Thiết kế quy trình Review**

**Ứng dụng quản lý sự kiện – Quy trình review tài liệu yêu cầu**

**a. Các bước thực hiện.**

1. **Lập kế hoạch**:
   * Xác định mục tiêu: Đảm bảo tài liệu yêu cầu rõ ràng, không mâu thuẫn, và đáp ứng nhu cầu của stakeholder.
   * Chọn loại review: Inspection (do yêu cầu cần kiểm tra kỹ lưỡng để đảm bảo chất lượng cao).
   * Phân công vai trò: Tác giả, người review (bao gồm đại diện đội phát triển, kiểm thử, và stakeholder), và người điều phối.
2. **Chuẩn bị**:
   * Gửi tài liệu yêu cầu cho người review ít nhất 3 ngày trước buổi họp.
   * Người review đọc trước, ghi chú các vấn đề tiềm ẩn (như yêu cầu không rõ ràng, mâu thuẫn).
3. **Thực hiện review**:
   * Tổ chức cuộc họp inspection, người điều phối dẫn dắt, tác giả giải thích nội dung tài liệu.
   * Người review trình bày các lỗi hoặc câu hỏi, thảo luận để thống nhất cách sửa.
4. **Báo cáo và sửa lỗi**:
   * Người điều phối tổng hợp danh sách lỗi, phân loại theo mức độ nghiêm trọng, và gửi cho tác giả.
   * Tác giả sửa chữa trong vòng 2 ngày và gửi lại tài liệu đã cập nhật.
5. **Kiểm tra lại**:
   * Người điều phối và người review kiểm tra tài liệu đã sửa, đảm bảo tất cả lỗi được khắc phục.

**b. Vai trò của các thành viên tham gia.**

1. **Tác giả**: Người viết tài liệu yêu cầu (ví dụ: nhà phân tích nghiệp vụ).
   * Trách nhiệm: Chuẩn bị tài liệu, giải thích các yêu cầu, sửa lỗi theo phản hồi.
2. **Người review**: Bao gồm lập trình viên, nhân viên kiểm thử, và stakeholder (người tổ chức sự kiện, người dùng cuối).
   * Trách nhiệm: Phân tích tài liệu, phát hiện lỗi hoặc thiếu sót, cung cấp phản hồi.
3. **Người điều phối**: Quản lý dự án hoặc trưởng nhóm QA.
   * Trách nhiệm: Điều hành cuộc họp, đảm bảo quy trình hiệu quả, theo dõi tiến độ sửa lỗi.

**c. Loại review được chọn và lý do.**

* **Loại review**: Inspection.
* **Lý do lựa chọn**:
  + Tài liệu yêu cầu là nền tảng cho toàn bộ dự án, cần đảm bảo chất lượng cao và không có lỗi nghiêm trọng.
  + Inspection là quy trình trang trọng, có cấu trúc rõ ràng, giúp phát hiện lỗi một cách kỹ lưỡng.
  + Phù hợp với dự án quản lý sự kiện, nơi yêu cầu cần rõ ràng để đáp ứng nhu cầu của nhiều stakeholder (người tổ chức, người tham gia).

#### **Phân tích vai trò của phản hồi từ Stakeholder**

**Dự án phát triển ứng dụng đặt lịch bác sĩ – Tài liệu thiết kế giao diện thiếu tùy chọn hủy lịch hẹn.**

**a. Phản hồi từ stakeholder giúp phát hiện vấn đề.**

* **Cách phát hiện**:
  + **Phản hồi từ bệnh nhân**: Trong buổi review, bệnh nhân chỉ ra rằng không có tùy chọn hủy lịch hẹn sẽ gây khó khăn nếu họ thay đổi kế hoạch, dẫn đến trải nghiệm người dùng kém.
  + **Phản hồi từ bác sĩ**: Bác sĩ yêu cầu cần có chức năng hủy lịch để quản lý lịch trình hiệu quả hơn, tránh trường hợp bệnh nhân không đến mà lịch vẫn bị chiếm.
  + **Kết quả**: Phản hồi này giúp phát hiện thiếu sót trong thiết kế giao diện, đảm bảo hệ thống đáp ứng nhu cầu thực tế của cả hai nhóm stakeholder.

**b. Kế hoạch thu thập và tích hợp phản hồi từ stakeholder.**

1. **Thu thập phản hồi**:
   * **Khảo sát hoặc phỏng vấn**: Gửi bảng câu hỏi hoặc tổ chức phỏng vấn với bệnh nhân và bác sĩ để thu thập ý kiến về thiết kế giao diện (ví dụ: các tính năng cần có, vấn đề tiềm ẩn).
   * **Buổi review với stakeholder**: Mời đại diện bệnh nhân và bác sĩ tham gia buổi review tài liệu thiết kế giao diện để đưa ra phản hồi trực tiếp.
   * **Dùng nguyên mẫu (prototype)**: Tạo một giao diện mẫu đơn giản và cho stakeholder trải nghiệm để xác định các tính năng còn thiếu (như nút hủy lịch).
2. **Tích hợp phản hồi**:
   * **Cập nhật tài liệu**: Thêm yêu cầu về tùy chọn hủy lịch hẹn vào tài liệu thiết kế, ví dụ: “Giao diện hiển thị nút ‘Hủy lịch’ bên cạnh mỗi lịch hẹn đã đặt, với thông báo xác nhận trước khi hủy.”
   * **Xác nhận với stakeholder**: Gửi tài liệu đã cập nhật cho stakeholder để xác nhận rằng các thay đổi đáp ứng nhu cầu của họ.
   * **Kiểm tra lại**: Tổ chức một buổi review thứ hai để đảm bảo tất cả phản hồi đã được tích hợp đúng và không có vấn đề mới phát sinh.
3. **Lợi ích**:
   * Cải thiện trải nghiệm người dùng bằng cách đáp ứng nhu cầu thực tế.
   * Tăng tính khả thi của hệ thống, đảm bảo cả bệnh nhân và bác sĩ có thể sử dụng hiệu quả.
   * Giảm nguy cơ phải sửa đổi giao diện ở giai đoạn sau, tiết kiệm chi phí và thời gian.