**Prompt 1: Unit testing trong kiểm thử phần mềm – Ý nghĩa, mục tiêu và ứng dụng thực tế**

**Bối cảnh:** Bạn là sinh viên đang tham gia một dự án nhóm phát triển phần mềm nhỏ để xây dựng ứng dụng quản lý thư viện cho trường Đại học. Sau khi hoàn thành việc học về các mức kiểm thử, bạn được giao nhiệm vụ lên kế hoạch kiểm thử rõ ràng, trong đó phải nêu chi tiết về từng mức kiểm thử sẽ được áp dụng trong dự án này. Trình bày ngắn gọn thử: Unit Testing. Nêu rõ mục đích, ý nghĩa và phạm vi áp dụng cụ thể.

**1. Unit Testing (Kiểm thử đơn vị)**

* **Mục đích & Ý nghĩa:**  
  Unit Testing tập trung kiểm thử các đơn vị nhỏ nhất của phần mềm như hàm, module, hoặc đối tượng (object) riêng biệt, nhằm xác nhận rằng từng đơn vị chức năng hoạt động đúng theo yêu cầu kỹ thuật đã được xác định. Theo sách, mục tiêu chính là tìm ra lỗi (defect) càng sớm càng tốt và đảm bảo rằng các thành phần nhỏ này không chứa lỗi trước khi tích hợp vào các phần lớn hơn của hệ thống.
* **Phạm vi áp dụng:**  
  Trong dự án quản lý thư viện, Unit Testing sẽ kiểm tra từng hàm riêng lẻ như hàm xác thực đăng nhập, hàm tìm kiếm sách theo mã, hoặc module tính phí phạt khi trả sách muộn.

**Prompt 2: Phân biệt Verification và Validation**

**Bối cảnh:** Bạn là thành viên của một nhóm sinh viên thực hiện dự án phát triển ứng dụng đặt phòng ký túc xá. Bạn nhận thấy mọi người thường dùng lẫn lộn hai khái niệm “verification” (xác minh) và “validation” (thẩm định). Bạn phải trình bày ngắn gọn, rõ ràng để mọi thành viên đều hiểu và áp dụng đúng hai khái niệm này trong các giai đoạn kiểm thử của dự án. Yêu cầu bạn phải định nghĩa rõ ràng verification và validation trong kiểm thử phần mềm. Nêu ra sự khác biệt cơ bản giữa hai hoạt động này. Đưa một ví dụ cụ thể cho mỗi khái niệm, gắn liền với chức năng trong dự án đặt phòng ký túc xá.

* **Verification (Xác minh):** Là quá trình kiểm tra xem sản phẩm có được xây dựng đúng theo đặc tả kỹ thuật, tài liệu thiết kế và tiêu chuẩn đặt ra hay không (“Are we building the product right?”). Verification thường gồm các hoạt động như review code, kiểm tra thiết kế, walkthrough.
* **Validation (Thẩm định):** Là quá trình kiểm tra để đảm bảo sản phẩm thực sự phù hợp với nhu cầu, mong đợi của người dùng cuối (“Are we building the right product?”). Validation thường bao gồm kiểm thử chấp nhận, demo và trải nghiệm thực tế.
* **Khác biệt cơ bản:**
  + Verification kiểm tra sự đúng đắn về kỹ thuật và thiết kế.
  + Validation kiểm tra sự phù hợp và hữu ích đối với người dùng cuối.
* **Ví dụ với ứng dụng đặt phòng ký túc xá:**
  + Verification: Nhóm kiểm tra chức năng đặt phòng đúng với tài liệu thiết kế, đúng logic kỹ thuật.
  + Validation: Sinh viên và quản lý ký túc xá sử dụng thử ứng dụng; nếu mọi người thấy tiện lợi, dễ dùng và đúng nhu cầu, validation được đáp ứng.

**Prompt 3: Acceptance Testing và rủi ro**

**Bối cảnh:** Sau khi học về các mức kiểm thử phần mềm, bạn nhận ra rằng mỗi mức kiểm thử đều có vai trò riêng trong việc phát hiện lỗi và giảm rủi ro cho dự án. Bạn muốn hiểu rõ hơn về điểm mạnh và điểm hạn chế của từng mức kiểm thử. Giải thích ngắn gọn Acceptance Testing và mô tả mục tiêu chính. Chú ý loại lỗi hoặc rủi ro nào thường được phát hiện tốt nhất ở mức đó? Nếu bỏ qua một mức kiểm thử bất kỳ, có thể dẫn đến hậu quả hoặc rủi ro gì cho sản phẩm phần mềm? Đưa ví dụ thực tế.

* **Acceptance Testing (Kiểm thử chấp nhận):** Là mức kiểm thử cuối cùng, nhằm xác nhận rằng phần mềm đáp ứng đúng các yêu cầu và kỳ vọng thực tế của khách hàng hoặc người dùng cuối. Mục tiêu chính là đảm bảo hệ thống thực sự sẵn sàng để triển khai sử dụng.
* **Loại lỗi/rủi ro phát hiện tốt:** Thường phát hiện các lỗi về thiếu sót chức năng, quy trình nghiệp vụ không phù hợp, hoặc giao diện gây khó khăn cho người dùng. Nếu bỏ qua Acceptance Testing, sản phẩm có thể đạt chuẩn kỹ thuật nhưng không đáp ứng nhu cầu thực tế, khiến khách hàng từ chối sử dụng hoặc phải sửa lại sau khi đã triển khai.
* **Ví dụ:** Trong ứng dụng đặt phòng ký túc xá, nếu không có acceptance testing, hệ thống có thể thiếu chức năng cho phép sinh viên xem trước danh sách phòng còn trống theo từng tòa nhà. Kết quả là sinh viên phải thử từng phòng một, gây mất thời gian.

**Prompt 4: So sánh Integration Testing và System Testing**

**Bối cảnh:** Bạn làm việc trong một dự án phát triển ứng dụng quản lý trường học. Nhóm bạn đã hoàn thành Unit Testing và chuẩn bị chuyển sang các mức kiểm thử tiếp theo. Trình bày ngắn gọn mục tiêu và phạm vi của Integration Testing và System Testing. So sánh điểm giống và khác nhau giữa hai mức kiểm thử này về mục đích, đối tượng kiểm thử, thời điểm thực hiện.

* **Integration Testing (Kiểm thử tích hợp):** Có mục tiêu kiểm tra sự tương tác giữa các module hoặc đơn vị phần mềm đã qua kiểm thử đơn vị, đảm bảo chúng hoạt động chính xác khi kết hợp với nhau. Phạm vi là kiểm thử các giao diện, luồng dữ liệu, sự trao đổi thông tin giữa các thành phần trong hệ thống (ví dụ: giữa module quản lý học sinh và module điểm danh). Integration testing giúp phát hiện lỗi ở ranh giới giữa các module, như lỗi truyền tham số, xử lý dữ liệu hoặc gọi hàm.
* **System Testing (Kiểm thử hệ thống):** Có mục tiêu kiểm tra toàn bộ phần mềm đã hoàn thiện như một thể thống nhất, dựa trên các yêu cầu chức năng và phi chức năng đã đặt ra. Phạm vi là toàn bộ hệ thống, đánh giá quy trình nghiệp vụ, hiệu năng, bảo mật, khả năng sử dụng. Đây là bước xác nhận phần mềm đáp ứng đúng yêu cầu đã xác định ban đầu, trước khi chuyển sang kiểm thử chấp nhận.
* **So sánh:**
  + Giống nhau: Đều nhằm phát hiện lỗi sau khi các thành phần riêng lẻ đã được kiểm thử (sau Unit Testing). Đều đảm bảo các thành phần phối hợp đúng để tạo ra hệ thống hoàn chỉnh.
  + Khác nhau:
    - Mục đích: Integration testing kiểm thử sự kết hợp giữa các module; system testing kiểm thử toàn bộ hệ thống.
    - Đối tượng kiểm thử: Integration testing kiểm thử các cụm module đã tích hợp; system testing kiểm thử cả hệ thống như một sản phẩm hoàn chỉnh.
    - Thời điểm: Integration testing sau unit testing, khi các module vừa được tích hợp; system testing sau khi toàn bộ hệ thống đã hoàn thiện.

**Prompt 5: Vai trò Acceptance Testing với khách hàng và người dùng cuối**

**Bối cảnh:** Sau khi nhóm bạn hoàn thành các mức kiểm thử trước (unit, integration, system), bạn chuẩn bị chuyển sang Acceptance Testing trước khi bàn giao phần mềm quản lý trường học cho nhà trường. Trình bày ngắn gọn mục tiêu chính của Acceptance Testing và ai là người thực hiện kiểm thử này. Phân tích Acceptance Testing giúp phát hiện ra những vấn đề gì mà các mức kiểm thử trước chưa thể nhận ra. Nêu ví dụ cụ thể về một yêu cầu của khách hàng/chức năng thực tế chỉ được phát hiện là chưa phù hợp trong quá trình Acceptance Testing của dự án trường học.

* **Acceptance Testing (Kiểm thử chấp nhận):** Là mức kiểm thử cuối cùng, với mục tiêu xác nhận phần mềm đáp ứng đúng các yêu cầu và mong đợi của khách hàng hoặc người dùng cuối trước khi triển khai sử dụng thực tế. Acceptance Testing thường được thực hiện bởi khách hàng, người dùng cuối, hoặc đại diện nhà trường.
* **Giúp phát hiện vấn đề:** Acceptance Testing giúp phát hiện các vấn đề mà các mức kiểm thử trước (unit, integration, system) chưa thể nhận ra, như: sản phẩm chưa phù hợp với quy trình làm việc thực tế; tính năng thiếu sót hoặc chưa đúng với nhu cầu thực tiễn; trải nghiệm sử dụng còn bất tiện với người dùng cuối.
* **Ví dụ thực tế:** Trong dự án quản lý trường học, chỉ đến giai đoạn Acceptance Testing, đại diện nhà trường mới phát hiện hệ thống điểm danh không có chức năng cho phép xuất danh sách điểm danh ra file Excel, trong khi đây là yêu cầu quan trọng để tổng hợp và báo cáo hàng tháng. Trước đó, nhóm phát triển không phát hiện ra thiếu sót này trong các mức kiểm thử trước.