**SOFTWARE PRODUCT QUALITY - ISO/IEC 25010**

## **1. Functional Suitability**

### **1.1 Functional completeness**

* **Ý nghĩa**: Mức độ mà sản phẩm cung cấp đầy đủ các chức năng cần thiết.
* **Ví dụ**: Ứng dụng ngân hàng có chức năng chuyển khoản, thanh toán hóa đơn, xem lịch sử giao dịch.
* **Validate**: So sánh yêu cầu nghiệp vụ với các chức năng đang triển khai; dùng checklist để kiểm tra độ đầy đủ.

### **1.2 Functional correctness**

* **Ý nghĩa**: Mức độ mà sản phẩm trả về kết quả đúng khi sử dụng.
* **Ví dụ**: Ứng dụng tính thuế VAT ra kết quả chính xác theo công thức pháp luật.
* **Validate**: Viết test case với đầu vào xác định, kiểm thử kết quả đầu ra đúng tuyệt đối.

### **1.3 Functional appropriateness**

* **Ý nghĩa**: Mức độ mà chức năng hỗ trợ tốt việc hoàn thành tác vụ.
* **Ví dụ**: Chức năng tự động gợi ý địa chỉ khi người dùng đặt hàng online.
* **Validate**: Quan sát quá trình sử dụng của người dùng (usability testing), phân tích hiệu quả hoàn thành nhiệm vụ.

## **2. Performance Efficiency**

### **2.1 Time behaviour**

* **Ý nghĩa**: Tốc độ phản hồi và xử lý của hệ thống.
* **Ví dụ**: Trang web tải trong vòng <3 giây với tốc độ mạng tiêu chuẩn.
* **Validate**: Dùng công cụ như JMeter, Lighthouse để đo thời gian phản hồi, TTFB, throughput.

### **2.2 Resource utilization**

* **Ý nghĩa**: Mức độ sử dụng tài nguyên (CPU, RAM, băng thông...).
* **Ví dụ**: App nền sử dụng <10% CPU trong trạng thái idle.
* **Validate**: Sử dụng công cụ đo hiệu suất hệ thống (Task Manager, top, perfmon...).

### **2.3 Capacity**

* **Ý nghĩa**: Giới hạn tối đa mà hệ thống có thể xử lý.
* **Ví dụ**: Hệ thống học trực tuyến phục vụ đồng thời 10.000 sinh viên.
* **Validate**: Kiểm thử tải (load testing, stress testing) với số lượng người dùng tăng dần.

## **3. Compatibility**

### **3.1 Co-existence**

* **Ý nghĩa**: Mức độ hệ thống hoạt động ổn định cùng các hệ thống khác.
* **Ví dụ**: Ứng dụng video call không xung đột với phần mềm diệt virus.
* **Validate**: Cài và chạy đồng thời trên môi trường thật, quan sát lỗi hoặc xung đột tài nguyên.

### **3.2 Interoperability**

* **Ý nghĩa**: Mức độ trao đổi và sử dụng thông tin với hệ thống khác.
* **Ví dụ**: Website bán hàng tích hợp với cổng thanh toán PayPal.
* **Validate**: Kiểm thử tích hợp (integration testing), gửi request/response qua API, SOAP/REST.

## **4. Interaction Capability**

### **4.1 Appropriateness recognizability**

* **Ý nghĩa**: Người dùng có dễ dàng nhận biết hệ thống phù hợp với nhu cầu không.
* **Ví dụ**: Trang chủ của ứng dụng học tiếng Anh hiển thị rõ các cấp độ học và mục tiêu.
* **Validate**: Khảo sát người dùng mới, phân tích hành vi qua heatmap/click tracking.

### **4.2 Learnability**

* **Ý nghĩa**: Mức độ dễ học để sử dụng hệ thống.
* **Ví dụ**: Người mới dùng thành thạo app đặt xe sau 1–2 lần thử.
* **Validate**: Thử nghiệm với người dùng mới, đo thời gian học và lỗi phát sinh.

### **4.3 Operability**

* **Ý nghĩa**: Mức độ dễ dàng vận hành hệ thống.
* **Ví dụ**: Menu dễ hiểu, biểu tượng nhất quán, hỗ trợ thao tác nhanh.
* **Validate**: Thử nghiệm người dùng; đo số lần thao tác trung bình cho 1 tác vụ.

### **4.4 User error protection**

* **Ý nghĩa**: Mức độ hệ thống giúp người dùng tránh lỗi.
* **Ví dụ**: Hệ thống cảnh báo khi người dùng nhập email sai định dạng.
* **Validate**: Xem các rule validation, kiểm thử hành vi khi nhập sai dữ liệu.

### **4.5 User engagement**

* **Ý nghĩa**: Mức độ giao diện khuyến khích người dùng tương tác tiếp tục.
* **Ví dụ**: Game học từ vựng có hình ảnh và phần thưởng hấp dẫn.
* **Validate**: Phân tích tỉ lệ retention, heatmap, A/B testing giao diện.

### **4.6 Inclusivity**

* **Ý nghĩa**: Hệ thống có thể phục vụ người dùng đa dạng về văn hóa, tuổi tác, năng lực.
* **Ví dụ**: App ngân hàng có giao diện hỗ trợ đa ngôn ngữ và phông chữ lớn cho người cao tuổi.
* **Validate**: Xem xét tính năng hỗ trợ đa ngôn ngữ, khả năng tùy chỉnh giao diện.

### **4.7 User assistance**

* **Ý nghĩa**: Mức độ hỗ trợ người dùng trong quá trình sử dụng.
* **Ví dụ**: Tooltip hướng dẫn, trợ lý ảo chatbot, tài liệu sử dụng tích hợp.
* **Validate**: Đánh giá mức độ hiện diện của hướng dẫn trong từng màn hình; khảo sát hỗ trợ.

### **4.8 Self-descriptiveness**

* **Ý nghĩa**: Mức độ hệ thống tự giải thích cách dùng mà không cần tài liệu ngoài.
* **Ví dụ**: Biểu tượng rõ ràng, hướng dẫn trực tiếp trên UI.
* **Validate**: Giao cho người mới thao tác thử; đánh giá xem họ có cần hỏi hướng dẫn ngoài không.

## **5.** **Reliability**

### **5.1 Faultlessness**

* **Ý nghĩa**: Hệ thống hoạt động chính xác, không gây lỗi dưới điều kiện bình thường.
* **Ví dụ**: App xử lý đơn hàng không bị lỗi khi đặt nhiều sản phẩm cùng lúc.
* **Validate**: Thực hiện kiểm thử chức năng với các tình huống sử dụng phổ biến.

### **5.2 Availability**

* **Ý nghĩa**: Mức độ hệ thống sẵn sàng để sử dụng bất kỳ lúc nào cần.
* **Ví dụ**: Dịch vụ ngân hàng trực tuyến hoạt động 24/7.
* **Validate**: Giám sát uptime qua các công cụ như Pingdom, New Relic; phân tích log downtime.

### **5.3 Fault tolerance**

* **Ý nghĩa**: Khả năng hệ thống tiếp tục hoạt động dù có lỗi phần mềm hoặc phần cứng.
* **Ví dụ**: Server phụ tự động nhận tải khi server chính gặp sự cố.
* **Validate**: Mô phỏng lỗi phần cứng hoặc phần mềm; đo khả năng tiếp tục xử lý.

### **5.4 Recoverability**

* **Ý nghĩa**: Hệ thống có khả năng phục hồi dữ liệu và trạng thái sau sự cố.
* **Ví dụ**: Ứng dụng có chức năng tự động khôi phục dữ liệu sau crash.
* **Validate**: Kiểm tra backup, restore; kiểm thử khả năng khôi phục sau khi ngắt kết nối hoặc tắt nguồn.

## **6. Security**

### **6.1 Confidentiality**

* **Ý nghĩa**: Mức độ hệ thống bảo vệ thông tin không bị truy cập trái phép.
* **Ví dụ**: Hệ thống lưu trữ mật khẩu dưới dạng đã mã hóa (hash + salt).
* **Validate**: Kiểm tra mã hóa dữ liệu, quyền truy cập, thử nghiệm xâm nhập (penetration testing).

### **6.2 Integrity**

* **Ý nghĩa**: Mức độ hệ thống bảo vệ dữ liệu khỏi bị thay đổi trái phép.
* **Ví dụ**: Hệ thống kiểm tra checksum để phát hiện dữ liệu bị thay đổi.
* **Validate**: Dùng các kỹ thuật kiểm tra tính toàn vẹn (checksum, hash), thử thay đổi dữ liệu qua API trái phép.

### **6.3 Non-repudiation**

* **Ý nghĩa**: Mức độ hệ thống ngăn người dùng từ chối hành động đã thực hiện.
* **Ví dụ**: Hệ thống ghi log chi tiết và ký số (digital signature) khi người dùng thực hiện giao dịch.
* **Validate**: Kiểm tra log, kiểm thử khả năng đối chiếu truy vết hành động của người dùng.

### **6.4 Accountability**

* **Ý nghĩa**: Mức độ hệ thống truy vết được hành vi người dùng.
* **Ví dụ**: Hệ thống ghi nhận userID, thời gian, hành động trong log bảo mật.
* **Validate**: Kiểm tra log audit; đảm bảo mỗi hành vi được gắn với một người dùng cụ thể.

### **6.5 Authenticity**

* **Ý nghĩa**: Mức độ xác minh danh tính người dùng hoặc hệ thống đúng đắn.
* **Ví dụ**: Yêu cầu đăng nhập bằng xác thực hai yếu tố (2FA).
* **Validate**: Kiểm thử cơ chế đăng nhập, xác minh token, đánh giá quy trình xác thực.

## **7. Maintainability**

### **7.1 Modularity**

* **Ý nghĩa**: Mức độ hệ thống được chia thành các mô-đun độc lập.
* **Ví dụ**: Hệ thống quản lý học tập có module riêng cho đăng ký, học liệu, bài kiểm tra.
* **Validate**: Đánh giá kiến trúc phần mềm, sự tách biệt module qua mã nguồn hoặc sơ đồ.

### **7.2 Reusability**

* **Ý nghĩa**: Mức độ một phần của hệ thống có thể tái sử dụng trong các hệ thống khác.
* **Ví dụ**: Thư viện tính toán biểu thức được dùng trong nhiều ứng dụng kế toán.
* **Validate**: Kiểm tra tính độc lập của module; mức độ trừu tượng và cấu trúc mã.

### **7.3 Analysability**

* **Ý nghĩa**: Mức độ dễ dàng phân tích lỗi, tác động thay đổi hoặc vấn đề hệ thống.
* **Ví dụ**: Log chi tiết giúp nhanh chóng xác định nguyên nhân lỗi hệ thống.
* **Validate**: Đọc log, kiểm tra tài liệu, kiểm tra khả năng debug.

### **7.4 Modifiability**

* **Ý nghĩa**: Mức độ dễ sửa đổi hệ thống mà không gây lỗi khác.
* **Ví dụ**: Sửa giao diện không làm ảnh hưởng đến chức năng xử lý dữ liệu.
* **Validate**: Đo effort khi thay đổi code, kiểm tra test regression.

### **7.5 Testability**

* **Ý nghĩa**: Mức độ dễ viết và thực hiện kiểm thử hệ thống.
* **Ví dụ**: Hệ thống chia thành các hàm nhỏ dễ viết unit test.
* **Validate**: Kiểm tra coverage của kiểm thử; mức độ hỗ trợ kiểm thử tự động.

## **8. Portability**

### **8.1 Adaptability**

* **Ý nghĩa**: Mức độ hệ thống thích ứng với môi trường hoặc nền tảng khác nhau.
* **Ví dụ**: Ứng dụng web hoạt động tốt cả trên desktop và mobile.
* **Validate**: Kiểm thử trên các môi trường phần cứng, hệ điều hành, trình duyệt khác nhau.

### **8.2 Scalability**

* **Ý nghĩa**: Khả năng mở rộng quy mô theo nhu cầu sử dụng.
* **Ví dụ**: Dịch vụ streaming có thể phục vụ hàng triệu người dùng đồng thời khi có sự kiện lớn.
* **Validate**: Kiểm thử hiệu năng với quy mô tăng dần; đo phản hồi, tải hệ thống.

### **8.3 Installability**

* **Ý nghĩa**: Mức độ dễ dàng cài đặt hoặc gỡ bỏ hệ thống.
* **Ví dụ**: Ứng dụng desktop có file cài đặt đơn giản, tự động phát hiện cấu hình.
* **Validate**: Kiểm thử quá trình cài đặt/gỡ cài đặt trên các môi trường.

### **8.4 Replaceability**

* **Ý nghĩa**: Mức độ hệ thống thay thế được hệ thống khác với chức năng tương đương.
* **Ví dụ**: Cài phần mềm kế toán mới thay thế phần mềm cũ nhưng vẫn tương thích với dữ liệu cũ.
* **Validate**: Thử chuyển đổi dữ liệu và sử dụng hệ thống mới trong cùng môi trường vận hành.

## **9. Safety**

### **9.1 Operational constraint**

* **Ý nghĩa**: Mức độ hệ thống đảm bảo giới hạn vận hành an toàn khi xảy ra nguy cơ.
* **Ví dụ**: Xe tự hành giảm tốc hoặc dừng khi phát hiện vật cản.
* **Validate**: Mô phỏng các tình huống nguy hiểm; kiểm tra phản ứng hệ thống khi vượt ngưỡng vận hành.

### **9.2 Risk identification**

* **Ý nghĩa**: Mức độ hệ thống có khả năng phát hiện rủi ro tiềm ẩn.
* **Ví dụ**: Phần mềm y tế cảnh báo liều thuốc vượt mức an toàn.
* **Validate**: Rà soát logic nhận diện rủi ro; thử nghiệm với dữ liệu giả lập có rủi ro.

### **9.3 Fail safe**

* **Ý nghĩa**: Mức độ hệ thống tự chuyển sang chế độ an toàn khi gặp lỗi.
* **Ví dụ**: Hệ thống điện tự ngắt khi phát hiện quá tải.
* **Validate**: Mô phỏng lỗi và quan sát hệ thống có chuyển sang chế độ an toàn hay không.

### **9.4 Hazard warning**

* **Ý nghĩa**: Mức độ hệ thống đưa ra cảnh báo khi phát hiện nguy cơ.
* **Ví dụ**: Ứng dụng máy công nghiệp báo động khi nhiệt độ vượt ngưỡng cho phép.
* **Validate**: Kiểm thử phản hồi cảnh báo; đánh giá thời gian phản ứng.

### **9.5 Safe integration**

* **Ý nghĩa**: Mức độ hệ thống duy trì an toàn khi tích hợp với hệ thống khác.
* **Ví dụ**: Thiết bị y tế vẫn hoạt động an toàn khi kết nối với hệ thống giám sát trung tâm.
* **Validate**: Kiểm thử tích hợp, mô phỏng tương tác; phân tích tác động đến an toàn khi kết nối thành phần mới.