

[illegible]

3	Thời gian đáp ứng (Response Time)	Đánh giá thời gian cần để hoàn thành một nhiệm vụ nhất định là bao nhiêu?	Công thức: T= A - B A: Thời điểm nhận được kết quả đo B: Thời điểm kết thúc vào lệnh	T đạt theo ngưỡng đạt của KPI chất lượng dịch vụ	Lấy theo từng dịch vụ theo bộ KPI chất lượng dịch vụ GP CNTT hoặc theo yêu cầu bài toán. - Đạt ngưỡng: 10 - Vượt ngưỡng nhỏ hơn hoặc bằng 30%: 5 - Vượt ngưỡng lớn hơn 30%: 0	- Chương trình - Công cụ kiểm thử - Kết quả mong muốn của khách hàng	Sử dụng công cụ kiểm thử hiệu năng và thực hiện đo load test. - Thực hiện kiểm thử các chức năng hoặc các bộ tích hợp các chức năng với số lượng CCU (Concurrent Users) mà bài toán yêu cầu (số lượng CCU đáp ứng mong muốn). - Đo tham số response time tương ứng của từng chức năng/ bộ tích hợp. - Công cụ thường dùng để đo: Loadrunner, JMeter.. - Trong trường hợp chưa có công cụ đo, đội dự án tự xây dựng công cụ kiểm thử cho ứng dụng của mình.	Kiểm thử	Trước nghiệm thu nội bộ	Trước triển khai cho KH	N/A	
4	Thông lượng (Throughput)	Đánh giá có bao nhiêu tác vụ được thực hiện thành công trong một thời gian nhất định?	$X = A / T$ A = Số lượng tác vụ được thực hiện thành công T = thời gian thực hiện đo.	Tất cả các tác vụ được thực hiện thành công trong khoảng thời gian yêu cầu.	Theo yêu cầu của từng bài toán. Trong đơn vị thời gian: - Tất cả các tác vụ được thực hiện thành công: 10 - Lớn hơn hoặc bằng 90% tác vụ thực hiện thành công: 8. - Nhỏ hơn 90% tác vụ được thực hiện thành công: 0	- Chương trình - Công cụ kiểm thử - Kết quả mong muốn của khách hàng	Sử dụng công cụ kiểm thử và thực hiện đo load test. - Thực hiện kiểm thử 1 chức năng/1 bộ tích hợp các chức năng với số lượng CCU (Concurrent Users) mà bài toán yêu cầu (số lượng CCU đáp ứng mong muốn). Định nghĩa: Hits/ second được định nghĩa là số http request nhận được thành công bởi VU (Virtual Users) - Công cụ thường dùng để đo: Loadrunner, JMeter.. - Trong trường hợp chưa có công cụ đo, đội dự án tự xây dựng công cụ kiểm thử cho ứng dụng của mình.	Kiểm thử	Trước nghiệm thu nội bộ	Trước triển khai cho KH	N/A	
5	Năng lực (Capacity Measure)	Đánh giá có bao nhiêu người dùng có thể truy cập ứng dụng tại cùng 1 thời điểm.	$X = A/T$ A: số lượng truy cập đồng thời lớn nhất ứng dụng đáp ứng được. T: thời điểm ứng dụng đáp ứng được số lượng truy cập đồng thời lớn nhất.	Số lượng truy cập thỏa mãn yêu cầu bài toán.	Theo yêu cầu của từng bài toán. - Số lượng truy cập bằng với yêu cầu: 10 - Số lượng truy cập >= 90% số lượng truy cập yêu cầu: 8 - Số lượng truy cập < 90% số lượng truy cập yêu cầu: 0	- Chương trình - Công cụ kiểm thử - Kết quả mong muốn của khách hàng	- Sử dụng công cụ kiểm thử và thực hiện đo stress test. + Thực hiện kiểm thử 1 chức năng hoặc 1 bộ tích hợp các chức năng chạy trong môi trường giả lập trong thời gian T1 nhất định. Tăng dần số CCU thực hiện chức năng/ bộ tích hợp đó cho đến khi server đạt điểm nghẽn/ chết. + Tại giới điểm ngay trước điểm ngưỡng làm chết server chính là thời điểm số lượng truy cập đồng thời lớn nhất mà ứng dụng đáp ứng được. Thông tin tham khảo: - Giá trị thường dùng của khoảng thời gian T1: 1 giờ. - Server chết: được định nghĩa bằng thống số response time lớn hơn yêu cầu và tỷ lệ fail lớn hơn 10%. - Công cụ thường dùng để đo: Loadrunner, JMeter.. - Trong trường hợp chưa có công cụ đo, đội dự án tự xây dựng công cụ kiểm thử cho ứng dụng của mình.	Kiểm thử	Trước nghiệm thu nội bộ	Trước triển khai cho KH	N/A	

3. Tính tương thích (Compatability)

Mục đích: đánh giá khả năng phần mềm/ hệ thống có thể trao đổi với hệ thống/ sản phẩm khác, hoặc thực hiện chức năng của nó trong khi chia sẻ chung môi trường phần cứng và phần mềm.

6	Khả năng cùng tồn tại (Co-existence metrics)	Đánh giá phần mềm có thể cùng tồn tại với phần mềm khác trong môi trường chung và chia sẻ tài nguyên chung. Như phần mềm có khả năng cài cùng application server với phần mềm khác hay không. Phần mềm có thể dùng chung database server với phần mềm khác hay không ?.	Tỷ lệ lỗi mà người dùng gặp phải khi phần mềm được triển khai trên cùng môi trường và chia sẻ tài nguyên dùng chung với phần mềm khác. Tài nguyên dùng chung tại trung tâm hiện bao gồm: 1/ Chung máy chủ ứng dụng 2/ Chung cơ sở dữ liệu 3/ Chung hệ thống xác thực người dùng	Đạt tất cả yêu cầu trong hướng dẫn thực hiện.	Đạt tất cả các yêu cầu trong hướng dẫn thực hiện: 10; Không đạt tiêu chí 1 trong hướng dẫn thực hiện: trừ 6 điểm; Không đạt tiêu chí 2 và 3 trong hướng dẫn thực hiện: mỗi tiêu chí trừ 2 điểm.	+ Chương trình	1/ Cấu hình số lượng kết nối CSDL tối đa cho mỗi ứng dụng để đảm bảo khi một hệ thống có quá nhiều kết nối tại thời điểm quá tải chiếm hết toàn bộ số lượng kết nối mà máy chủ cơ sở dữ liệu có thể cung cấp làm ảnh hưởng đến ứng dụng khác được cài đặt cùng CSDL; (yêu cầu quan trọng hơn) 2/ Người vận hành khai thác kiểm tra xem có xảy ra hiện tượng tranh chấp tài nguyên, tranh chấp cổng... hay không; 3/ Cấu hình dung lượng RAM tối đa mà ứng dụng có thể sử dụng để đảm bảo không chiếm toàn bộ RAM của máy chủ ứng dụng làm ảnh hưởng đến các ứng dụng được cài đặt chung khác.	Giải pháp, Lập trình, Kiểm thử, Triển khai	Trước khi chuyển giao sang đơn vị VHKT	Trước khi chuyển giao sang đơn vị VHKT	NA	x
7	Khả năng liên kết (Interoperability metrics)	Xác định khả năng tương tác với một hoặc một vài hệ thống cụ thể của phần mềm	X= 1 - A / B A= Số lượng các trường hợp trong đó trao đổi giữa các hệ thống bị failed. B= Tổng số trường hợp trao đổi dữ liệu mong đợi thành công. Các đội dự án đưa ra một process để đo sự thành công trong trao đổi dữ liệu giữa các hệ thống (không tập trung vào độ chính xác dữ liệu mà tập trung vào sự trao đổi dữ liệu) Đo thời gian để trao đổi dữ liệu/ xử lý một yêu cầu có đáp ứng được mục tiêu trong mô tả hay không? Áp dụng cho hệ thống có tương tác/ giao tiếp với hệ thống khác. Lưu ý: chỉ tập trung vào đầu vào và đầu ra khi trao đổi giữa các hệ thống, không quan tâm đến nghiệp vụ người dùng.	Tất cả các tps đều đạt so với yêu cầu	Tất cả các tps đều đạt so với yêu cầu: 10 Tất cả các tps đều đạt 90% so với yêu cầu: 5 Có ít nhất một tps không đạt 90% so với yêu cầu: 0	PTYC/ MTYCTĐ/ DSYC Bộ dữ liệu mẫu và kết quả mong muốn của khách hàng Tool giả lập request Bảo cáo kiểm thử Sản phẩm hoàn thiện	Giao tiếp giữa các hệ thống có 2 loại: tuân theo chuẩn, hoặc không tuân thủ theo chuẩn (ví dụ hệ thống BCCS, KTTS, ERP Tập đoàn). Khi kiểm tra giao tiếp với các hệ thống cần xem xét các tiêu chí sau: 1) Thông tin trao đổi có đúng định dạng như đã định nghĩa hay không? 2) Các xử lý ngoại lệ như khi hệ thống đích timeout, hệ thống đích gửi về các thông báo lỗi, mất kết nối có được xử lý đầy đủ không? 3) Thời gian thực hiện việc trao đổi dữ liệu có đúng như yêu cầu của bài toán không? Đo tính chính xác của việc trao đổi dữ liệu đã được kiểm tra ở tính phù hợp chức năng (yếu tố 1 và 2). Ở đây chỉ tập trung vào đo thời gian giao tiếp giữa các hệ thống. Cách thức thực hiện đo: - Mỗi ứng dụng định nghĩa thời gian trao đổi của các ứng dụng (ví dụ đồng bộ dữ liệu thông qua Golden gate có mục tiêu trao đổi là: 20record/s) - Ghi log trong quá trình viết tiến trình trao đổi dữ liệu giữa các hệ thống (tại thời điểm bắt đầu thực hiện và thời điểm kết thúc quá trình). - Đọc từ log và tính toán đơn vị thời gian giao tiếp giữa các hệ thống (ví dụ: đồng bộ được 200 record/ 2 phút) - So sánh kết quả tính toán thời gian giao tiếp và mục tiêu mong muốn ở trên -> giao tiếp failed nếu kết quả đo thực > mục tiêu mong muốn ở trên	Giải pháp, Lập trình, Kiểm thử, Triển khai	Trước nghiệm thu nội bộ	Trước triển khai cho KH	N/A	
4. Tính dễ sử dụng (Usability)												
Mục đích: Đánh giá sự dễ hiểu, dễ học, khả năng vận hành và tính thẩm mỹ của phần mềm bằng các tài liệu hướng dẫn và những quy tắc/ quy định trong phát triển phần mềm.												
8	Tính dễ hiểu (Understandability)	Đánh giá mức độ dễ hiểu của phần mềm đối với người dùng.	Chọn mẫu 10 - 20% luồng chức năng chính 1/ Mức độ dễ hiểu của mô tả chức năng: thời gian đọc hiểu chức năng thông qua mô tả chức năng càng ngắn càng tốt. 2/ Mức độ tiện dụng của hệ thống trợ giúp: càng chi tiết, trực quan càng tốt. 3/ Mức độ dễ hiểu của chức năng: thời gian tìm hiểu chức năng thông qua thao tác trực tiếp càng ngắn càng tốt.	Đạt tất cả yêu cầu trong hướng dẫn thực hiện.	Đạt tất cả các yêu cầu trong hướng dẫn thực hiện: 10; Không đạt tiêu chí 1 trong hướng dẫn thực hiện: trừ 4 điểm; Không đạt tiêu chí 2,3,4 trong hướng dẫn thực hiện: trừ 2 điểm;	- HDSD, kịch bản sử dụng, tài liệu mô tả sử dụng, HD VHKT, HD CD. - Chương trình	Khi xây dựng/thiết kế phần mềm để đảm bảo tính dễ hiểu thì cần thực hiện như sau: 1. Tài liệu hướng dẫn sử dụng chi tiết, rõ ràng, trình bày trực quan, sinh động, có ví dụ cụ thể. 2. Các thuật ngữ trên giao diện (Label, Button, Menu ...) phải chuẩn, ngắn gọn và dễ hiểu. 3. Có các thông báo trạng thái đang xử lý của phần mềm. 4. Dữ liệu đầu vào và đầu ra đảm bảo người dùng hiểu được dễ dàng.	Giải pháp, Lập trình, Kiểm thử	Trước triển khai cho KH	Trước triển khai cho KH	NA	x

9	Tính dễ học (Learnability)	Đánh giá mức độ dễ học của phần mềm đối với người dùng.	Chọn mẫu 10 - 20% luồng chức năng chính 1/ Mức độ dễ học chức năng. 2/ Mức độ hiệu quả của hệ thống trợ giúp và hướng dẫn.	Đạt tất cả yêu cầu trong hướng dẫn thực hiện.	Đạt tất cả các yêu cầu trong hướng dẫn thực hiện: 10; Không đạt tiêu chí trong hướng dẫn thực hiện: mỗi tiêu chí trừ 2.5 điểm;	- Chương trình	Khi xây dựng/thiết kế phần mềm để đảm bảo tính dễ học thì cần thực hiện như sau: 1/ Hệ thống menu được phân cấp rõ ràng, logic. 2/ Trong mỗi menu nên có icon để giúp người dùng dễ nhận diện chức năng. 3/ Hệ thống có sử dụng Tool Tips để người dùng dễ hiểu 4/ Hệ thống có tính năng Help(F1) để người dùng có thể tra cứu nếu không biết sử dụng.	Giải pháp, Lập trình, Kiểm thử	Trước triển khai cho KH	Trước triển khai cho KH	NA	x
10	Khả năng vận hành (Operability)	Đánh giá mức độ dễ thao tác của phần mềm đối với người dùng.	Chọn mẫu 10 - 20% luồng chức năng chính 1/ Khả năng làm chủ hệ thống: người quản trị có thể dễ dàng sửa được lỗi của họ khi thao tác. 2/ Mức độ rõ ràng của thông báo: thông báo lỗi, hướng dẫn phải rõ ràng trong quá trình người quản trị vận hành ứng dụng. 3/ Khả năng chống lỗi, khôi phục khi có lỗi. 4/ Khả năng hỗ trợ người dùng nhập liệu: người dùng có thể dễ dàng lựa chọn giá trị cần nhập, có thể thay đổi tham số.	Đạt tất cả yêu cầu trong hướng dẫn thực hiện.	Đạt tất cả các tiêu chí bắt buộc trong hướng dẫn thực hiện: 10; Không đạt các tiêu chí bắt buộc trong hướng dẫn thực hiện: mỗi tiêu chí trừ 2 điểm; Đạt tiêu chí không bắt buộc cộng 1 điểm với 1 tiêu chí, nhưng bảo đảm điểm tổng không vượt quá 10 điểm.	- Chương trình	Một phần mềm được đánh giá là dễ thao tác đối với người dùng khi có những thuộc tính sau: Tiêu chí bắt buộc với tất cả các loại ứng dụng: + Khi vào các chức năng có các giá trị mặc định. + Có chức năng cấu hình/ file cấu hình giúp người dùng dễ thay đổi cấu hình để phù hợp với yêu cầu. + Các thông báo phải rõ nghĩa khi có lỗi xảy ra. + Dữ liệu người dùng nhập vào và có hướng dẫn nhập dữ liệu đối với các loại dữ liệu cần nhập đúng định dạng như ngày tháng, tiền,... + Có các phím tắt Tiêu chí không bắt buộc: + Mỗi thao tác quan trọng cần có chức năng undo giúp người dùng khôi phục lại trạng thái ban đầu khi thao tác sai.	Giải pháp, Lập trình, Kiểm thử	Trước triển khai cho KH	Trước triển khai cho KH	NA	X
11	Thẩm mỹ của giao diện (User interface aesthetics)	Đánh giá giao diện có thể thỏa mãn các nhu cầu tương tác của người dùng.	1/ Khả năng người dùng có thể sửa đổi (customize) giao diện theo sở thích của mình. 2/ Sự hài hòa của giao diện thể hiện ở bố cục, sự cân đối, màu sắc, kết cấu, luồng màn hình, đồ họa.	Đạt tất cả yêu cầu trong hướng dẫn thực hiện.	Đạt tất cả các yêu cầu trong hướng dẫn thực hiện: 10; Không đạt tiêu chí trong hướng dẫn thực hiện: trừ 2.5 điểm;	- Chương trình	1/ Khi thiết kế giao diện cần tuân thủ các nguyên tắc sau: 1.1) Đồ họa, màu sắc, font chữ: thống nhất giao diện. + Hình ảnh và các icons trên giao diện phải được thiết kế theo cùng phong cách trên toàn ứng dụng (trên tất cả các màn hình). + Đồ họa, màu sắc, font chữ cần được phối hợp hài hòa và đồng nhất. 1.2) Tỷ lệ và vị trí đặt của hình ảnh: + Kích cỡ và vị trí của hình ảnh cần được đặt tương ứng với độ quan trọng của nó. + Gom nhóm thông tin và collapse thông tin theo nhóm	Giải pháp, Lập trình, Kiểm thử	- Giai đoạn thiết kế giao diện, thiết kế chương trình. - Trước triển khai cho KH.	- Giai đoạn thiết kế giao diện, thiết kế chương trình. - Trước triển khai cho KH.		X
5. Tính tin cậy (Reliability)												
Mục đích: Đánh giá khả năng phần mềm có thể hoạt động tin cậy trong những điều kiện cụ thể (hoàn thiện về sản phẩm)												
12	Độ bao phủ kiểm thử (Test Coverage)	Đánh giá mức độ đầy đủ của testcase qua đó đánh giá sản phẩm có đủ tin cậy trước khi triển khai hay không	X = A/ (A+B) A: Số lượng Testcases thiếu sau review B: Tổng số TCs đã viết	100%	X = 0: 10 0% <X< 10% : 8 10% <= X <= 30% : 5 X > 30% : 0	PTYC/DTYC/Tài liệu thiết kế, KBKT	Người đánh giá xem xét ít nhất 10% các testcases trong KBKT tương ứng với từng chức năng.	Giải pháp, Kiểm thử	Trước nghiệm thu nội bộ	Trước triển khai cho KH	Càng nhỏ càng tốt	

13	Lỗi tiềm ẩn chức năng (Potential Fault)	Đánh giá lỗi tiềm ẩn bao gồm lỗi không tuân thủ quy tắc lập trình có thể phát sinh thành lỗi chức năng khi triển khai và sử dụng	Đánh giá lỗi tiềm ẩn bao gồm lỗi không tuân thủ quy tắc lập trình có thể phát sinh thành lỗi chức năng khi triển khai và sử dụng. (Không/ khó được phát hiện bởi quá trình kiểm thử). Yêu cầu khắc phục hết tất cả các lỗi tiềm ẩn phát hiện trong chương trình. □	Fix hết tất cả các lỗi tiềm ẩn	Fix hết tất cả các lỗi tiềm ẩn: 10, Không fix hết tất cả các lỗi: 0	+ Mã nguồn	<ul style="list-style-type: none"> - Số lỗi tiềm ẩn chưa khắc phục lấy từ kết quả quá lỗi tiềm ẩn của chương trình bằng công cụ quét. Nếu chưa có công cụ quét thì check ngẫu nhiên ít nhất 10% số chức năng. Danh sách các lỗi tiềm ẩn cần phải khắc phục: 1. Tuân thủ coding convention đã ban hành của Trung tâm 2. Fix hết các cảnh báo của IDE lập trình 3. Lỗi dữ liệu 4. Lỗi cấu trúc điều khiển 5. Lỗi nhập xuất 6. Lỗi giao tiếp giữa các hàm, thủ tục 7. Lỗi sử dụng bộ nhớ (Tham khảo phụ lục Checklist Lỗi tiềm ẩn)	Lập trình	Trước nghiệm thu nội bộ	Trước triển khai cho KH	Càng nhỏ càng tốt	
14	Lỗi tiềm ẩn hiệu năng (Potential Fault)	Đánh giá lỗi tiềm ẩn bao gồm lỗi không tuân thủ quy tắc lập trình có thể phát sinh thành lỗi chức năng khi triển khai và sử dụng	Đánh giá lỗi tiềm ẩn bao gồm lỗi không tuân thủ quy tắc lập trình có thể phát sinh thành lỗi hiệu năng khi triển khai và sử dụng. (Không/ khó được phát hiện bởi quá trình kiểm thử). Yêu cầu khắc phục hết tất cả các lỗi tiềm ẩn phát hiện trong chương trình.	Fix hết tất cả các lỗi tiềm ẩn	Fix hết tất cả các lỗi tiềm ẩn: 10, Không fix hết tất cả các lỗi: 0	Mã nguồn	Phần này tập trung vào phần lập trình SQL. Các tiêu chí cần kiểm tra bao gồm: 1/ Thực hiện theo hướng dẫn viết câu lệnh SQL (thực hiện lệnh select như quy định, giảm thiểu số lượng subsequencies trong truy vấn, sử dụng exist thích hợp...) 2/ Tuân thủ theo hướng dẫn đánh Index (loại dữ liệu nào, tần suất đánh Index...) 3/ Tuân thủ theo hướng dẫn đánh Partitioning (loại dữ liệu thực hiện partitioning, hướng dẫn đánh partitioning với từng loại dữ liệu...) (Tham khảo phụ lục Checklist Lập trình SQL)	Lập trình	Trước nghiệm thu nội bộ	Trước triển khai cho KH	Càng nhỏ càng tốt	

6. Tính an toàn (Security)

Mục đích: xác định khả năng bảo vệ thông tin và dữ liệu của sản phẩm phần mềm, sao cho người, hệ thống không được phép thì không thể truy cập, đọc hay chỉnh sửa chúng.

15	Tính an toàn (Security metrics)	Xác định khả năng bảo vệ thông tin và dữ liệu của sản phẩm phần mềm, sao cho người, hệ thống không được phép thì không thể truy cập, đọc hay chỉnh sửa chúng.	<p>Đánh giá dựa theo hướng dẫn lập trình an toàn trong phát triển ứng dụng web, iOS, android được ban hành của Tập đoàn. Các lỗi được mô tả vắn tắt như sau:</p> <ul style="list-style-type: none"> - Lỗ hổng SQL injection, lỗ hổng XSS, lỗ hổng CSRF, kiểm soát các thao tác với file, mã hóa dữ liệu nhạy cảm, quyền truy cập của người dùng, user enumeration, session fixation, session hijacking, HTTP only cookie, chuyển hướng chuyển tiếp thiếu thẩm tra, dễ lộ dữ liệu của hệ thống, thất thoát thông tin do kiểm soát lỗi và ngoại lệ không tốt. - Ngoài ra bổ sung tính accountability: lập trình để đảm bảo có thể truy vết lại từng thao tác. - Với các ứng dụng sử dụng chữ ký số lưu ý tính non certification. Tức là sau khi thử nghiệm 	Pass qua kiểm tra ATTT	Không còn lỗi ATTT: 10 Còn lỗi ATTT: 0	Báo cáo đánh giá ATTT Sản phẩm hoàn thiện	<ul style="list-style-type: none"> - Trong quá trình phát triển, đội phát triển tuân thủ theo hướng dẫn lập trình ATTT của Tập đoàn và xem xét nội bộ. - Trước nghiệm thu nội bộ, dự án gửi yêu cầu kiểm tra ATTT lên đơn vị kiểm tra cuối cấp Trung tâm. - Nếu phát hiện ra lỗi thì phải sửa hết trước khi gửi lại đơn vị kiểm tra trước triển khai cho KH. + Chú ý: không có ngoại lệ nếu còn lỗi ATTT, tức là tất cả các lỗi ATTT đều phải được sửa trước khi triển khai cho KH. Thống số này là bắt buộc. + Các lưu ý về tính accountability, chữ ký số chưa phải yêu cầu bắt buộc thời điểm này. 	Lập trình, Kiểm thử	Trước nghiệm thu nội bộ	Trước triển khai cho KH	Không còn lỗi ATTT
----	---------------------------------	---	---	------------------------	---	---	--	---------------------	-------------------------	-------------------------	--------------------

7. Khả năng bảo trì (Maintainability)

Mục đích: Xác định khả năng có thể chỉnh sửa được của phần mềm. Việc chỉnh sửa bao gồm: sửa lại cho đúng, cải tiến, thêm mới làm phần mềm thích nghi được với những thay đổi của môi trường, của yêu cầu và của chức năng xác định.

16	Phân tích được (Analysability metrics)	Đánh giá mức độ đầy đủ của việc ghi log	Đánh giá mức độ đầy đủ của việc ghi log phục vụ việc phân tích, xử lý khi có lỗi xảy ra, phát hiện các điểm treo của hệ thống.	Đạt tất cả yêu cầu trong hướng dẫn thực hiện.	Đạt tất cả các yêu cầu trong hướng dẫn thực hiện: 10; Không đạt tiêu chí 1, 2, 3, 4 trong hướng dẫn thực hiện: mỗi tiêu chí trừ 1.75 điểm; Không đạt tiêu chí 5, 6, 7 trong hướng dẫn thực hiện: mỗi tiêu chí trừ 1 điểm;	+ Mã nguồn + Tài liệu Quy định ghi log (chưa ban hành)	Dùng công cụ kiểm tra. Khi chưa có công cụ thì thực hiện xem xét một số file mã nguồn để đánh giá. Kiểm tra ít nhất 10% chức năng nghiệp vụ chính 1/ Tài liệu PTYC phải mô tả đầy đủ thông tin về ghi log. 2/ Đã ghi log những tác động của người dùng chưa? 3/ Nội dung log đã đúng theo mô tả trong tài liệu PTYC chưa? 4/ Hệ thống có dữ liệu quan trọng không? Đã ghi log tác động dữ liệu quan trọng chưa? 5/ Đã ghi log đúng quy chuẩn chưa? 6/ Đã ghi log cho từng phương thức chưa? 7/ Đã ghi log cho các lệnh lặp, rẽ nhánh ... chưa? (Tham khảo phụ lục Checklist ghi log)	Giải pháp, Lập trình, Kiểm thử	Trước nghiệm thu nội bộ	Trước triển khai cho KH	Không còn lỗi ATTT	x
17	Tính dễ kiểm thử (Testability metrics)	Đánh giá mức độ kiểm thử dễ dàng của phần mềm. Từ đó khi có một chỉnh sửa có thể dễ dàng đánh giá được sản phẩm.	Một phần mềm được xem là kiểm thử được khi có những thuộc tính sau: 1/ Cung cấp đầy đủ các công cụ/chức năng để kiểm thử; 2/ Kiểm thử tự động được; 3/ Tài liệu để kiểm thử rõ ràng dễ hiểu	Đạt tất cả yêu cầu trong hướng dẫn thực hiện.	Đạt tất cả các yêu cầu trong hướng dẫn thực hiện: 10; Không đạt tiêu chí 1, 2, 3 trong hướng dẫn thực hiện: mỗi tiêu chí trừ 2 điểm; Không đạt tiêu chí 4, 5, 6 trong hướng dẫn thực hiện: mỗi tiêu chí 1.3 điểm. Lưu ý: Với tiêu chí 5, 6 là tiêu chí dành cho hệ thống đặc thù, hệ thống nào không cần đến, nếu không có thì không bị trừ điểm. Tiêu chí không bắt buộc, nếu có cộng 1 điểm, nhưng bảo đảm điểm tổng không quá 10 điểm	+ Chương trình	Tiêu chí bắt buộc với tất cả các loại ứng dụng: 1/ Tài liệu giải pháp mô tả đúng và đủ các phần thay đổi, ảnh hưởng để kiểm thử lại khi có yêu cầu sửa đổi phần mềm. 2/ Tài liệu giải pháp, thiết kế phải chi tiết, rõ ràng và đầy đủ 3/ KBKT mô tả đầy đủ để hiểu các bước để người thực hiện có thể thực hiện chính xác. 4/ Trong quá trình lập trình dùng các công cụ kiểm thử tự động như JUnit; 5/ Có các công cụ mô phỏng/giả lập để kiểm thử cho các chức năng liên quan đến thiết bị, môi trường, hệ thống khác; 6/ Khi xây dựng chương trình bổ sung các chức năng để tự kiểm thử hệ thống (Ví dụ: trong hệ thống tính cước bổ sung các chức năng tính thử, ...). Tiêu chí không bắt buộc với tất cả các loại ứng dụng: 7/ Trong quá trình kiểm thử nghiệm cứu và áp dụng các công cụ kiểm thử như IBM Rational Functional Test, ...	Giải pháp, Lập trình, Kiểm thử	Trước nghiệm thu nội bộ	Trước triển khai cho KH	NA	X
18	Thay đổi được (Changeability metrics)	Đánh giá khả năng tương thích với các thay đổi của người dùng	1/ Đánh giá khả năng thay đổi của hệ thống thông qua cấu hình mà phải lập trình thêm; 2/ Đánh giá khả năng thay đổi của hệ thống để có thể khắc phục lỗi; 3/ Đánh giá khả năng thay đổi của hệ thống để tương thích với 1 yêu cầu mới. 4/ Dễ dàng tăng số lượng người dùng 5/ Có thể tăng số lượng thread, số lượng apps.	Đạt tất cả yêu cầu trong hướng dẫn thực hiện.	Đạt tất cả các yêu cầu trong hướng dẫn thực hiện: 10; Không đạt tiêu chí 1,2 trong hướng dẫn thực hiện: trừ 3 điểm; Không đạt tiêu chí 3,4 trong hướng dẫn thực hiện: trừ 2 điểm; Trường hợp hệ thống không yêu cầu bắt buộc tiêu chí 4 thì không trừ điểm	+ Chương trình	1/Hệ thống đã được phân tầng rõ ràng không? (Hệ thống có tăng xử lý nghiệp vụ riêng không để có thể hạn chế việc chỉnh sửa nếu chỉ sửa giao diện, có thể tái sử dụng lại được khi thêm một chức năng mới). 2/ Không fix các tham số cấu hình trong code mà lưu vào Database hoặc file để có thể sửa được nhanh chóng. 3/ Hệ thống có chuẩn hóa các hàm dùng chung thường xuyên sử dụng hay không (Một số hàm xử lý chung như chuyển kiểu ngày tháng, các hàm lấy dữ liệu hay dùng, ... để khi thêm một chức năng mới có thể dùng không phải viết lại). 4/ Hỗ trợ người dùng các chức năng cấu hình hệ thống (các chức năng phần mềm) để chỉnh sửa các tham số hệ thống qua các chức năng này mà không phải sửa trực tiếp trong CSDL hoặc file cấu hình.	Giải pháp, Lập trình, Kiểm thử	Trước nghiệm thu nội bộ	Trước triển khai cho KH	NA	x

8.Tính khả chuyển (Portability)

Mục đích: đánh giá khả năng cho phép của phần mềm có thể được chuyển từ môi trường này sang môi trường khác.

19	Khả năng tương thích (Adaptability metrics)	Đánh giá mức độ dễ chuyển đổi môi trường của phần mềm.	Áp dụng cho các hệ thống đòi hỏi tính realtime hoặc cần cài đặt trên hệ thống máy chủ khác nhau. Một phần mềm được xem là chuyển đổi được thì hệ thống có các thuộc tính sau: 1/ Chạy trên được nhiều cơ sở dữ liệu khác nhau như Oracle, DB2, MySQL, SQL Server ... 2/ Chạy được trên nhiều hệ điều hành khác như Windows Server, Oracle Solaris, Red hat, Windows XP, Ubuntu ... 3/ Chạy được trên nhiều thiết bị/phần cứng khác nhau như IBM, HP, SUN, DELL, ... 4/ Chuyển đổi được với sự thay đổi trong công ty/tổ chức. Khi có sự thay đổi trong nội bộ công ty/tổ chức như thay đổi mô hình kinh doanh, công nghệ, .. thì phần mềm vẫn đáp ứng được thay đổi đó.	Đạt tất cả yêu cầu trong hướng dẫn thực hiện.	Đạt tất cả các yêu cầu trong hướng dẫn thực hiện: 10; Không đạt tiêu chí 1, 2, 4, 4 trong hướng dẫn thực hiện: mỗi tiêu chí trừ 2.5 điểm;	+ Chương trình	1/ Trong quá trình phát triển lựa chọn các công nghệ độc lập với cơ sở dữ liệu như Hibernate, ... 2/ Trong quá trình phát triển lựa chọn các công nghệ/nền tảng phát triển không phụ thuộc vào môi trường của hệ điều hành như Java, ... 3/ Trong quá trình phát triển lựa chọn các công nghệ/nền tảng phát triển không phụ thuộc vào thiết bị/phần cứng 4/ Phần mềm thiết kế động hỗ trợ người dùng cấu hình và không fix cứng.	Giải pháp, Lập trình, Kiểm thử, Triển khai	Sau giai đoạn giải pháp, Thiết kế, đánh giá trên tài liệu: + Thiết kế tổng thể + Tài liệu mô tả sản phẩm	Trước triển khai cho KH	NA	x
----	--	--	--	--	---	----------------	---	--	--	-------------------------------------	----	---